

UNIVERSITÀ CA' FOSCARI VENEZIA



Facoltà di Lettere e Filosofia Corso di Laurea Specialistica in Informatica per le Discipline Umanistiche

Un ILS open source per l'automazione delle biblioteche.
L'ipotesi Koha a Ca' Foscari.

Relatore: Prof. Renzo Orsini

Correlatore: Prof. Riccardo Ridi

Laureando: Giulio Bonanome

Matricola: 818532

Anno accademico 2008-2009

*“Non hai veramente capito qualcosa
finché non sei in grado di spiegarlo a tua nonna.”*

Albert Einstein

Indice

1. Introduzione.....	6
2. L'automazione delle biblioteche.....	9
2.1. Perché automatizzare.....	9
2.2. Breve storia dell'automazione bibliotecaria.....	11
2.2.1. L'ottimizzazione del flusso di lavoro interno.....	12
2.2.2. L'accesso alle risorse locali.....	14
2.2.3. L'accesso alle risorse esterne.....	15
2.3. Strumenti per l'integrazione.....	17
2.4. Verso nuovi paradigmi.....	23
3. Integrated Library System.....	26
3.1. Premessa.....	26
3.2. Breve evoluzione degli ILS.....	27
3.3. Principali funzionalità di un ILS.....	28
3.3.1. Modulo per l'acquisizione.....	28
3.3.2. Modulo per la catalogazione.....	30
3.3.3. Modulo per il catalogo online.....	31
3.3.4. Modulo per la circolazione.....	31
3.3.5. Modulo per la gestione dei periodici.....	32
3.4. La dis-integrazione dei sistemi integrati.....	33
3.5. Criteri di selezione di un sistema integrato.....	36
3.6. Ultime tendenze e possibili sviluppi futuri.....	38
4. Software open source per l'automazione bibliotecaria.....	40
4.1. L'approccio della "sorgente aperta"	40
4.1.1. Free software, open source, FLOSS.....	41
4.1.2. Open source tra mito e realtà.....	46
4.2. Open source e biblioteche.....	48
4.2.1. ILS open source.....	53
4.2.2. Luci e ombre dei sistemi integrati open source.....	58

4.2.3. Criteri di valutazione di un ILS open source.....	59
5. La situazione in Italia.....	61
5.1. La lunga via della cooperazione.....	61
5.2. I sistemi integrati in Italia.....	69
5.3. Il software open source in Italia.....	74
6. Koha.....	77
6.1. Nascita ed evoluzione del software.....	77
6.2. Requisiti del sistema.....	79
6.3. L'installazione di Koha.....	81
6.4. La struttura del software.....	82
6.5. Struttura del database.....	86
6.5.1. Tabelle del modulo per le acquisizioni.....	90
6.5.2. Tabelle del modulo per la catalogazione.....	93
6.5.3. Tabelle del modulo per la circolazione.....	100
6.5.4. Tabelle del modulo per la gestione utenti.....	105
6.5.5. Tabelle del modulo per la gestione dei periodici.....	109
6.5.6. Tabelle relative ai reports.....	112
6.5.7. Tabelle di sistema.....	113
6.6. La comunità e gli strumenti di sviluppo.....	121
7. L'ipotesi Koha a Ca' Foscari.....	123
7.1. Il contesto di riferimento.....	123
7.2. L'analisi dei flussi operativi automatizzabili.....	125
7.2.1. Metodo d'indagine.....	126
7.2.2. Flusso operativo per le acquisizioni.....	128
7.2.3. Flusso operativo per la catalogazione.....	130
7.2.4. Flusso operativo per la gestione dei periodici.....	131
7.2.5. Flusso operativo per la circolazione.....	132
7.3. Considerazioni per l'adeguamento del software.....	133
7.4. Suggerimenti per un'ipotesi di migrazione.....	135
8. Conclusioni.....	137

Appendice A.....	141
Bibliografia.....	149
Siti web di riferimento.....	162
Ringraziamenti.....	164

Indice delle figure

Fig. 1 - Insieme di servizi e attori in un sistema bibliotecario.....	10
Fig. 2 - Moduli presenti in un sistema integrato.....	28
Fig. 3 - Esempi di comunicazione con il modulo acquisizioni.....	29
Fig. 4 - Esempi di comunicazione con il modulo catalogazione.....	30
Fig. 5 - Esempi di comunicazione con il catalogo online.....	31
Fig. 6 - Esempi di comunicazione con il modulo circolazione.....	32
Fig. 7 - Esempi di comunicazione con il modulo periodici.....	33
Fig. 8 - Sistema LAMP.....	79
Fig. 9 - Struttura a livelli di Koha.....	83
Fig. 10 - Modello E-R generalizzato per Koha 3.00.02	87
Fig. 11 - Modello E-R relativo al modulo per le acquisizioni.....	89
Fig. 12 - Modello E-R relativo al modulo per la catalogazione.....	92
Fig. 13 - Modello E-R relativo al modulo per la circolazione.....	99
Fig. 14 - Modello E-R relativo al modulo per la gestione utenti.....	104
Fig. 15 - Modello E-R relativo al modulo per i periodici.....	108
Fig. 16 - Modello E-R relativo ai reports.....	111
Fig. 17 - Workflow acquisizioni.....	128
Fig. 18 - Workflow catalogazione.....	130
Fig. 19 - Workflow periodici.....	131
Fig. 20 - Workflow circolazione.....	132

1. Introduzione

L'oggetto del presente lavoro è lo studio del sistema integrato open source Koha e l'ipotesi di una sua adozione da parte del Sistema Bibliotecario d'Ateneo di Ca' Foscari. L'obiettivo vorrebbe essere portare all'attenzione della comunità bibliotecaria le criticità legate all'adozione di sistemi integrati di questo tipo, sia per quanto riguarda l'adattamento alla realtà dei sistemi bibliotecari italiani, sia per quanto riguarda la diffusione di studi legati al tema dell'open source in questo ambito.

Per questo si è scelto di dedicare una prima parte del lavoro alla presentazione dei vari aspetti coinvolti: automazione bibliotecaria e sistemi integrati, open source, realtà italiana.

Per quanto riguarda l'automazione bibliotecaria, ovvero l'utilizzo di strumenti informatici all'interno di sistemi informativi bibliotecari, dopo una rapida panoramica circa l'evoluzione dei processi di automazione, verranno presi in considerazione gli ultimi sviluppi nel settore alla luce della diffusione di Internet e del web.

Gli Integrated Library Systems (ILS) sono uno degli strumenti principali dell'automazione bibliotecaria, dopo averne descritto i componenti principali e il modo in cui sono integrati tra loro, verranno presentate le attuali criticità nella selezione e le ultime tendenze di sviluppo.

Nell'affrontare il tema dell'open source invece ci si è scontrati innanzitutto con le difficoltà di definizione di un termine tanto usato in ambito informatico quanto poco familiare al di fuori di tale ambiente e per questo spesso vittima di fraintendimenti¹. In questo senso verranno fornite precisazioni circa la sua nascita ed evoluzione, per poi focalizzarsi sulla sua introduzione in ambito bibliotecario, soprattutto per quanto riguarda i sistemi integrati di questo tipo.

Infine verrà analizzata l'evoluzione dei processi d'automazione in Italia, con particolare attenzione alla storia del Servizio Bibliotecario Nazionale (SBN) e all'attuale panorama nell'offerta di sistemi integrati, nonché alla diffusione di software open source, soprattutto per quanto riguarda Pubblica Amministrazione e biblioteche.

Nella seconda parte del lavoro, dopo aver dettagliato la struttura di Koha, verrà fornita un'ipotesi di implementazione da parte del Sistema Bibliotecario d'Ateneo.

Per quanto riguarda la presentazione del sistema integrato Koha (versione 3.0.2), oltre alla descrizione del progetto e della comunità di riferimento, verranno analizzate la struttura e le funzionalità del sistema, focalizzandosi soprattutto sulla struttura del database MySQL.

L'ipotesi di utilizzo a Ca' Foscari consisterà in una rapida ricognizione del Sistema Bibliotecario, per poi fornire una descrizione più dettagliata dei flussi di lavoro direttamente collegati alle funzionalità del sistema integrato, prendendo come caso tipo una specifica biblioteca del sistema. In questo senso ci si è limitati all'analisi di quelle procedure automatizzabili dall'attuale versione di Koha, così come viene distribuita dalla comunità, consapevoli delle implementazioni necessarie prima di poterne offrire l'utilizzo in produzione. Infine verranno fatte delle proposte per l'adeguamento del software sulla base delle criticità rilevate e una scaletta delle operazioni necessarie ad un'eventuale migrazione.

Al termine del lavoro si proveranno ad evidenziare sia le problematiche relative

¹ A questo proposito, durante la presente trattazione, dove non diversamente specificato, i termini open source e software libero saranno utilizzati come sinonimi.

all'implementazione di un sistema integrato open source da parte di un Sistema Bibliotecario d'Ateneo, sia alcune considerazioni che potrebbero motivarne la scelta.

La speranza riposta in questo lavoro è che possa, da un lato, offrire una panoramica sul tema sufficientemente accessibile anche in assenza di profonde competenze informatiche, dall'altro, risultare un utile strumento nelle sperimentazioni di utilizzo di Koha.

2. L'automazione delle biblioteche

2.1. *Perché automatizzare*

Per fare riferimento all'applicazione dell'informatica al mondo delle biblioteche si parla comunemente di automazione bibliotecaria, indicando quali procedure o servizi si intende automatizzare e in che modo, ponendo l'accento di volta in volta sul tipo di tecnologie utilizzate oppure su come applicarle. Molto spesso, tuttavia, non si risponde a una domanda fondamentale: *perché automatizzare?*

Rispondendo a questa domanda, qualsiasi processo di automazione dovrebbe confrontarsi innanzitutto con gli obiettivi della biblioteca e inserirsi all'interno di un piano strategico che vede l'automazione come parte di un insieme più complesso di procedure. Queste sono orientate all'utente e si possono considerare come strutturate su più livelli, sulla base della tipologia di servizio coinvolto. Inoltre, nel compiere queste funzioni, una biblioteca può entrare in contatto con diversi attori, come: fornitori di materiale di vario tipo (p.e. libri, software), altre biblioteche (p.e. per il prestito interbibliotecario, come partecipanti a un consorzio), centri amministrativi (p.e. dipartimenti universitari, Ministeri).

In questo contesto, potremmo definire un sistema di automazione come quel sistema informatico di supporto al sistema informativo bibliotecario, inteso come insieme di elementi interconnessi che raccolgono o ricercano, elaborano, memorizzano e distribuiscono informazioni per supportare le attività gestionali,

decisionali e di controllo della biblioteca e per produrre valore agli utenti².

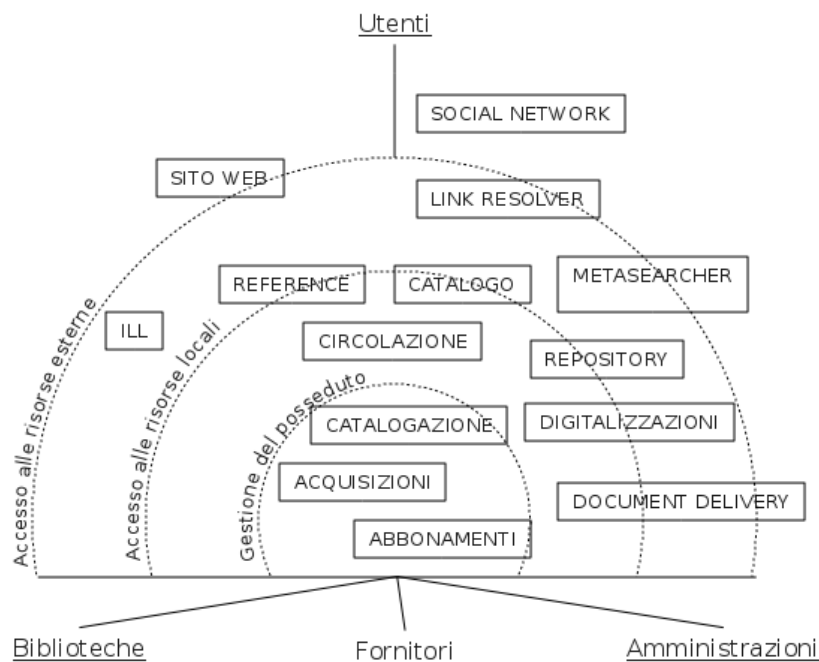


Figura 1: Insieme di servizi e attori in un sistema bibliotecario

I motivi principali che potrebbero spingere una biblioteca all'automazione sono i seguenti³:

- aumentare l'efficienza operativa;
- sollevare lo staff da mansioni impiegate per poterlo utilizzare in servizi orientati all'utente;
- migliorare la qualità, rapidità ed efficienza dei servizi;
- aumentare le possibilità di accesso da parte di utenti remoti o altri utenti interessati;
- migliorare l'accesso a risorse remote;
- fornire nuovi servizi;
- migliorare la gestione delle risorse fisiche e finanziarie;

2 Cfr. Leombroni [2007 p. 250]. Già Baldacci [1983] definiva "sistema informativo automatizzato [...] l'insieme delle apparecchiature fisiche, del software di base, dei programmi applicativi e dei dati. [...] uno strumento per migliorare il lavoro piuttosto che per automatizzarlo". Cfr. Baldacci [1983 p. 71].

3 Cfr. Haravu [2004 p. 26-27].

- facilitare un ampio accesso all'informazione per i loro utenti;
- facilitare un'ampia diffusione dei prodotti informativi e dei servizi della biblioteca;
- rendere possibile la partecipazione a reti di cooperazione tra biblioteche;
- aumentare la rapidità di comunicazione con altre biblioteche o istituzioni simili.

Queste motivazioni non sono mutualmente esclusive, anzi, spesso i vantaggi dell'automazione hanno un effetto a cascata, per cui una biblioteca consapevole di aver tratto vantaggi dalle applicazioni informatiche cercherà altre possibili aree di interesse da automatizzare, cercando sempre di condividere le scelte con l'istituzione e la comunità che serve.

Dal punto di vista delle tecnologie informatiche coinvolte, tradizionalmente, un sistema di automazione implementa almeno una delle seguenti caratteristiche:

- un sistema per la gestione del database (DBMS, database management system), utilizzato per creare e mantenere database bibliografici, garantendone la consistenza e l'accuratezza, anche attraverso la connessione all'authority file, e in grado di supportare gli standard internazionali relativi alla descrizione bibliografica e allo scambio di record bibliografici;
- sistemi per l'information retrieval, come ad esempio sistemi per l'indicizzazione o per la ricerca e visualizzazione dei record;
- architettura client-server, per la connessione dei calcolatori a una rete;
- applicativi web-based;

2.2. Breve storia dell'automazione bibliotecaria

Come condiviso da diversi studiosi⁴, l'evoluzione storica dell'automazione

⁴ Cfr. Borgman [1997], Lynch [2000]. E' importante segnalare che la seguente suddivisione storica fa riferimento soprattutto alla realtà anglosassone (Stati Uniti e Gran Bretagna). Ciononostante le principali motivazioni che hanno spinto, o spingono, all'automazione (rendere efficienti le

bibliotecaria può essere divisa in tre fasi principali, durante le quali ci si è concentrati principalmente sui seguenti obiettivi: ottimizzare le operazioni interne, favorire l'accesso alle risorse locali, favorire l'accesso alle risorse esterne.

2.2.1. L'ottimizzazione del flusso di lavoro interno

Durante la prima fase, tra gli anni '60 e gli anni '70, ci si è concentrati sul rendere efficienti le operazioni interne grazie all'utilizzo dei primi strumenti informatici.

Nei primi anni '60, con l'aumentare dei fondi per le collezioni bibliotecarie e l'espansione del settore dell'istruzione universitaria, aumentò rapidamente la quantità di pubblicazioni che le biblioteche dovevano gestire. Ben presto ci si rese conto che, non potendo gestire con la stessa rapidità di prima tale mole di materiale, era possibile automatizzare quelle operazioni più dispendiose e ripetitive, ottimizzando il flusso di lavoro.

Tuttavia automatizzare non era molto economico, così i primissimi progetti furono sviluppati dalle biblioteche stesse, esclusivamente per l'insieme di attività principali: circolazione, acquisizione, catalogazione e seriali.

Poiché in questo contesto catalogare restava una delle attività più dispendiose, molte biblioteche intuirono l'importanza della cooperazione e la realizzarono attraverso la condivisione dei dati bibliografici. In questo modo, in fase di catalogazione era possibile cercare il dato relativo all'oggetto da catalogare tra record già creati da altre biblioteche partner, oppure correggere il proprio record una volta scoperto che ne esisteva uno qualitativamente migliore, il tutto con un notevole risparmio in termini di tempo e margini di errore.

Uno degli esempi più celebri di catalogo condiviso online fu quello dell'Ohio College Library Center (OCLC, poi diventato Online Computer Library Center) nato

operazioni interne, accesso alle risorse locali della biblioteca, accesso alle risorse esterne presenti sulla rete), sono sostanzialmente le stesse anche nel resto del mondo e dove sono riscontrabili lentezze nell'attuazione di tale processo di automazione, potrebbero essere imputate a diverse politiche di investimento, oppure, altrettanto spesso, a problemi legati alla cooperazione.

nel 1967 con l'obiettivo della condivisione delle risorse e della riduzione dei costi⁵, a cui a breve giro, sempre negli Stati Uniti, seguirono il Research Libraries Information Network (RLIN) e il Western Library Network (WLN), in Inghilterra il Consortium of University Research Library (CURL), in Olanda il Project on Integrated Catalogue Automation (PICA).

Tutte queste esperienze di cooperazione furono rese possibili anche grazie all'avvio del progetto per la definizione di uno standard per lo scambio di dati bibliografici in un formato comprensibile dai computer. Nel 1963, infatti, la Library of Congress iniziò il suo lavoro su tale formato, arrivando nel 1966 alla definizione di un progetto pilota per l'utilizzo del MACHine-Readable Cataloging (MARC)⁶. Il lavoro per la definizione del nuovo formato MARC II venne portato avanti assieme alla British Library e portò, nel 1968, alle due versioni LC-MARC II negli Stati Uniti e BNB-MARC II in Inghilterra. La positiva esperienza spinse diversi paesi a sviluppare formati MARC nazionali⁷. Il MARC II gettò inoltre le basi per la definizione dello standard ISO 2709 (*Format for bibliographic information interchange on magnetic tape*) che fu pubblicato nel 1973.

In quegli stessi anni iniziò anche il lavoro da parte dell'International Federation of Library Association (IFLA) per la definizione di uno standard internazionale per la descrizione bibliografica (ISBD)⁸.

Tra gli anni '60 e la fine degli anni '70 la cooperazione tra le biblioteche, con la condivisione dei dati bibliografici, diventò il principale fattore che permise una rapida evoluzione del processo di automazione.

5 Attualmente OCLC è la rete bibliotecaria più ampia del mondo, con circa "27000 membri in 86 paesi" e un patrimonio di "768 milioni di libri e altro materiale catalogato e accessibile online". Cfr. OCLC [2009].

6 Per una concisa ma dettagliata introduzione alla storia e alla struttura del formato MARC cfr. Scolari [2000].

7 A titolo d'esempio cito solamente il francese MONOCLE nel 1970, che divenne INTERMARC nel 1975 e si estese al Belgio; nel 1973 il tedesco MAB I; nel 1977 l'italiano ANNAMARC. Accanto ai formati nazionali iniziò anche un lavoro per la creazione di uno standard internazionale che portò nel 1977 alla pubblicazione della definizione di UNIMARC. Quest'ultimo è attualmente il formato di scambio maggiormente diffuso nei paesi non anglosassoni, dove invece è privilegiato il MARC21.

8 La prima definizione di ISBD(M) è del 1974. L'importanza di ISBD è legata tanto alla sua rapida diffusione a livello internazionale quanto alla decisione di usarlo per la parte descrittiva della registrazione UNIMARC.

2.2.2. L'accesso alle risorse locali

La seconda fase del processo di automazione, tra gli anni '80 e i primi anni '90, si occupò di favorire l'accesso alle risorse locali della biblioteca.

Innanzitutto in quel periodo il costo dei computer diminuì notevolmente, permettendone una più ampia diffusione. In secondo luogo, la nascita degli integrated library systems (ILS) permise alle biblioteche di creare i record bibliografici direttamente al momento dell'ordinazione dei materiali, arricchendoli poi con le informazioni di acquisizione e catalogazione, ottimizzando ulteriormente il flusso di lavoro. La diffusione degli ILS favorì inoltre la nascita di un vero e proprio mercato dell'automazione bibliotecaria⁹.

Sempre nei primi anni '80, grazie alle prime reti locali (tra biblioteche o nei campus universitari), il catalogo venne messo online, diventando l'Online Public Access Catalog (OPAC) e permettendo così un più agevole accesso al catalogo da parte degli utenti. Uno dei vantaggi consisteva nel fatto che, mentre solitamente in un sistema bibliotecario formato da varie biblioteche, il catalogo unico era conservato esclusivamente presso l'istituto centrale e ogni biblioteca aveva in loco solo quello del suo posseduto, con l'OPAC era possibile l'accesso al catalogo attraverso i terminali presenti nelle varie biblioteche e collegati alla rete locale. Oltre al vantaggio di recuperare le informazioni attraverso un unico accesso, ora gli utenti avevano anche la possibilità di utilizzare strumenti come la ricerca per parole chiave, gli operatori booleani, filtri per data o tipo di materiale.

Arrivati a questo punto, per concretizzare i benefici dell'automazione, molte biblioteche iniziarono una conversione retrospettiva dei loro dati bibliografici, per evitare i costi di mantenimento di entrambi i cataloghi, cartaceo ed elettronico. Ovviamente fu un processo lungo e costoso, non solo per la quantità di dati da

⁹ Per avere una veloce idea dell'andamento del mercato per l'automazione bibliotecaria cfr. Breeding [2007c]

trattare, ma per il fatto che vi era la necessità di normalizzarli sulla base degli standard relativi alla descrizione bibliografica. A tutt'oggi non si può dire che questo processo sia stato completato e siamo ancora lontani dal controllo bibliografico universale¹⁰.

Questo insieme di attività portò nei primi anni '80 alla nascita di un vero e proprio mercato dell'automazione bibliotecaria, con la tendenza da parte delle biblioteche a passare dallo sviluppo di soluzioni interne, basate spesso su sistemi proprietari, all'acquisto di prodotti commerciali, possibilmente basati su standard tecnici.

La possibilità di ottenere informazioni sulla copia, anche a distanza, grazie ai cataloghi condivisi e all'OPAC, incentivò il prestito interbibliotecario (Interlibrary Loan, ILL) e il document delivery, resi ancora più efficienti dalla successiva nascita di Internet e del web.

2.2.3. L'accesso alle risorse esterne

I servizi di ILL e document delivery sono alcuni degli elementi caratterizzanti la terza fase della storia dell'automazione delle biblioteche. Negli anni '90, infatti, a fronte di un aumento esponenziale della produzione di documenti, le biblioteche hanno dovuto trovare nuovi modi per mantenere un servizio di qualità rivolgendo l'attenzione all'accesso alle risorse esterne presenti sulla rete.

La rapida diffusione delle risorse elettroniche¹¹, da un lato ha portato molte

10 "Con l'espressione «controllo bibliografico universale» si designa un sistema internazionale finalizzato a rendere universalmente e immediatamente disponibili, su diversi supporti e in una forma standardizzata definita e accettata in ambito internazionale, le registrazioni bibliografiche delle pubblicazioni prodotte in tutti i paesi; tale sistema si fonda sull'insieme dei sistemi nazionali di controllo bibliografico, responsabili, ciascuno per il proprio paese, della documentazione completa e autorevole della propria produzione editoriale." Solimine [1995].

11 Con risorsa elettronica si intende un "qualsiasi strumento disponibile tramite dispositivi elettronici in differenti formati di codifica; tuttavia nell'uso corrente l'accezione si restringe alla modalità digitale" ovvero "istruzioni, programmi e dati, memorizzati in formato digitale (ossia sotto forma di uno o più file) su un supporto esterno o interno di un elaboratore" Gambari – Guerrini [2002 p. 34-35]. In ambito biblioteconomico si distingue particolarmente tra risorse elettroniche locali (p.e. cd-rom, dvd) e risorse elettroniche remote (p.e. e-journal, banche dati (bibliografiche o a testo completo), e-book).

biblioteche a cercare di integrarne la gestione all'interno delle loro attività, dall'altro ha spinto a campagne di digitalizzazione, soprattutto da parte degli istituti in possesso di fondi "speciali", i quali, essendo spesso esenti da limitazioni dovute al copyright poiché di proprietà dell'istituzione stessa, sono diventati accessibili attraverso la rete¹².

Nasce il concetto di biblioteca digitale¹³, definita come un "un insieme organizzato di risorse, strumenti, servizi e personale specializzato, a cui una o più organizzazioni offrono prontamente e economicamente l'accesso, interpretano e distribuiscono le informazioni e assicurano la persistenza nel tempo delle collezioni digitali" [Tammaro, 2006].

Vengono potenziati gli strumenti che consentono di identificare, localizzare e ottenere i documenti grazie all'utilizzo di metadati e del protocollo Z39.50.

Con metadati si intendono, in senso molto ampio, dei "dati sui dati", ovvero un insieme di informazioni, generalmente strutturate secondo uno schema, relative a un qualche tipo di risorsa d'interesse; sono "costruiti, costruttivi e processabili"¹⁴ [Coyle 2010] e possono essere associati a vari tipi di risorse, soprattutto in ambito elettronico. Ad esempio, metadati descrittivi identificano la risorsa e forniscono informazioni sul suo contenuto, metadati amministrativi sono usati per aiutare alla gestione della risorsa¹⁵, metadati tecnici possono fornire informazioni sul tipo di file o sul formato o su altre caratteristiche tecniche¹⁶.

In ambito bibliotecario è nel 1995 che inizia un importante lavoro da parte dell'OCLC insieme al National Center for Supercomputing Applications (NCSA) per

12 Cfr. Lynch [2000 p.65].

13 A questo proposito è utile richiamare il quadrante della virtualizzazione proposto da Ridi [2007 p. 53], dove viene evidenziato come, all'aumento dell'automazione dei servizi e della quantità di materiale digitale gestito nella collezione, si accompagna una maggiore virtualizzazione della biblioteca. Nel tempo sono state accostate svariate definizioni a questo processo: biblioteca virtuale, biblioteca ibrida, biblioteca in rete, biblioteca online, biblioteca ibrida. Cfr. Gambari [2007]. Si veda anche AIB-GBD [2005].

14 Cfr. Coyle [2010]. "Costruiti" perché non esistono in natura, sono qualcosa di artificiale, "costruttivi" perché vengono creati per un qualche scopo, attività o per risolvere un qualche problema, "processabili" nel senso di utilizzabili in qualche modo, è importante poter agire sui metadati per riuscire a soddisfare le proprie esigenze.

15 Un esempio italiano sono i metadati amministrativi gestionali (MAG) promossi dall'ICCU nell'ambito del progetto Biblioteca Digitale Italiana. Cfr Ridi [2007], ICCU [2009].

16 Cfr. Hodgson [2002 p. 36].

la definizione di uno schema di metadati in grado di descrivere le risorse elettroniche presenti sul web. Nasce così il Dublin Core¹⁷, uno schema di 15 elementi in grado di fornire una descrizione minima di qualsiasi risorsa informativa digitale allo scopo di identificarla e permetterne la localizzazione e visualizzazione. A questi elementi sono poi associabili una serie di qualificatori in grado di specificarne il significato. I metadati possono essere collegati al documento in diversi modi: includendoli direttamente nel file oppure in un file separato richiamato da un link¹⁸. Il protocollo Z39.50¹⁹, basato sull'architettura client/server, standardizza i messaggi utilizzati tra client e server per il recupero di informazioni da uno o più database, a prescindere dalla piattaforma o software utilizzati. Nel 1988 viene approvato come standard ANSI/NISO, mentre nel 1995 ne viene pubblicata la terza versione, adottata successivamente anche dall'ISO (ISO-23950).

2.3. Strumenti per l'integrazione

Attualmente possiamo dire di essere entrati in una quarta fase, dove la parola chiave è integrazione.

Dal semplice scambio di dati bibliografici si passa all'interoperabilità dei sistemi, e le biblioteche oltre a fornire l'accesso alla propria collezione "offline", cercano di diventare punto di accesso al docuverso²⁰, inteso come l'insieme di tutti i documenti esistenti.

Proprio l'ottica dell'integrazione rende difficile delineare un quadro omogeneo delle innovazioni tecnologiche che hanno coinvolto le biblioteche nell'ultimo decennio. Innanzitutto perché in alcuni casi si è trattato di integrazioni con standard o tecnologie provenienti da diversi ambiti, successivamente perché alcune di queste

17 Attualmente il Dublin Core è diventato uno standard ANSI/NISO Z39.85-2007 e ISO-15836:2009.

18 Cfr. Benvenuti [2003]. Per il mantenimento dello schema di metadati e per rispondere in maniera adeguata alle nuove esigenze date dall'evoluzione della rete, è stata anche creata un'apposita organizzazione, la Dublin Core Metadata Initiative (DMCI).

19 Cfr. Hodgson [2002 p. 28].

20 Termine coniato negli anni '80 da Theodor Holm Nelson . Cfr. Ridi [2007 p.15].

innovazioni hanno generato forti dibattiti sugli strumenti fin'ora utilizzati dalle biblioteche, che in questa fase devono confrontarsi con i nuovi attori del mondo dell'informazione.

Innanzitutto, dal punto di vista degli standard, ricopre un ruolo centrale l'Extensible Markup Language (XML) che, pubblicato come raccomandazione ufficiale nel 1998 da parte del World Wide Web Consortium (W3C)²¹, offre un set di regole standard per la rappresentazione di strutture di dati. In sostanza si tratta di un meta-linguaggio di marcatura²² che permette di definire altri linguaggi e documenti. Il vantaggio di XML, oltre all'elevato supporto di UNICODE per la codifica dei caratteri²³, consiste nell'estensibilità, per cui può essere utilizzato nei contesti più diversi²⁴. Inoltre, essendo nato specificamente per il web, è lo standard utilizzato nella definizione di Web services²⁵.

La gestione delle risorse elettroniche è diventata una parte così importante nell'attività della biblioteca, da non poter più essere gestita esclusivamente attraverso i sistemi integrati, che non riescono a supportare adeguatamente il workflow operativo legato alla loro selezione, acquisizione e gestione²⁶; vengono perciò sviluppati software specifici, denominati Electronic Resource Management

21 Il Consorzio, fondato nel 1994 da Tim-Berners Lee, comprende 356 organizzazioni membri e ha come obiettivo lo sviluppo e il mantenimento di standard per il World Wide Web.

22 Un linguaggio di marcatura (es. HTML) permette l'annotazione di un testo e definisce i meccanismi di rappresentazione delle parti annotate. XML è un linguaggio di tipo descrittivo, poiché lascia la scelta del tipo di rappresentazione da applicare al testo al software che di volta in volta lo riprodurrà.

23 "I computer, in buona sostanza, non fanno altro che trattare numeri. Per immagazzinare in memoria lettere o altri segni è necessario che a ogni carattere venga assegnato un numero. Esistono centinaia di sistemi di codifica, preesistenti a Unicode, e ognuno di questi abbina i numeri ai caratteri in modo differente. Nessuno di queste codifiche comprende un numero di caratteri sufficiente per tutte le circostanze [...] Questi sistemi di codifica, inoltre, sono in contraddizione l'uno con l'altro. Succede che due codifiche utilizzino lo stesso numero per due caratteri *diversi* o che, viceversa, adottino numeri diversi per lo *stesso* carattere. [...] Il problema è che, quando i dati passano da una codifica a un'altra, o da una piattaforma a un'altra, si corre il serio rischio di perdere informazioni. Unicode attribuisce un numero univoco a ogni carattere, indipendentemente dalla piattaforma, dall'applicativo, dalla lingua." Cfr. UNICODE [2008].

24 Cfr. Cover [2005] dove sono elencati ben 594 progetti e iniziative basate su XML.

25 Un "servizio Web" è un sistema software progettato per supportare l'interoperabilità tra diversi elaboratori attraverso una rete. Ha un'interfaccia descritta in un formato comprensibile dalla macchina. Gli altri sistemi interagiscono con il Web service utilizzando messaggi SOAP (un protocollo per lo scambio di messaggi tra software), tipicamente veicolati usando il protocollo HTTP e formattati usando XML, in congiunzione con altri standard del web. Cfr. WS Arch-WG [2004].

26 Cfr. Breeding [2004]; Cassella – Gargiulo [2006].

Systems (ERMS). Questi applicativi, che possono essere integrati all'interno dell'ILS o essere utilizzati come applicazioni stand-alone, forniscono da un lato un'insieme di funzionalità per la gestione delle risorse elettroniche da parte dello staff (p.e. licenze, sottoscrizioni di e-journal), dall'altro strumenti utili agli utenti per l'accesso ai documenti (p.e. link resolver, link al full-text, ricerche federate, link al catalogo)²⁷. Per quanto riguarda gli strumenti di ricerca, vengono implementati sistemi per la metaricerca e sistemi per la ricerca federata, che permettono l'interrogazione simultanea di più risorse remote attraverso l'uso di un'unica interfaccia²⁸.

La necessità di localizzare le risorse in ambiente ipertestuale, porta allo sviluppo di servizi di reference linking, o linking citazionale, ovvero strumenti in grado di portare all'oggetto informativo partendo dalle informazioni contenute nella citazione bibliografica. Questo avviene grazie allo sviluppo dello schema OpenURL, un protocollo standard²⁹, nato in ambito accademico, per la localizzazione delle risorse attraverso lo scambio di metadati. Lo schema permette, partendo da una risorsa formattata secondo la sintassi OpenURL, e passando attraverso un software (*link resolver*) in grado di interpretare i metadati che la accompagnano, di arrivare a un menu di servizi contestuale, basato sulle disponibilità fornite dall'istituzione che ha prodotto la risorsa di partenza, che potranno andare dal full-text della risorsa citata alla possibilità di richiederne il prestito interbibliotecario³⁰. Per l'istituzione questo vuol dire fornire un punto di accesso unico alle varie risorse elettroniche (a pagamento o ad accesso libero) che compongono la sua collezione.

Oltre alla localizzazione della risorsa è comunque necessaria la sua identificazione, e a questo proposito è stato creato il Digital Object Identifier (DOI)³¹, un identificatore

27 Cfr. Breeding [2004].

28 A questo proposito, anche se dal punto di vista dell'utente non sembra esserci differenza, è bene chiarire la distinzione tra i due termini. Una "metaricerca" può essere definita un processo "just-in-time", dove la query fornita dall'utente viene sottoposta a diverse risorse nel momento in cui viene effettuata. Una "ricerca federata" è invece un processo "just-in-case", dove le risorse informative sono precedentemente incorporate in un unico *repository*. Cfr. Sadeh [2006].

29 Lo schema, il cui progetto venne iniziato tra il 1998 e il 2000 da Herbert Van de Sompel presso l'Università di Ghent, nel 2004 è diventato standard ANSI/NISO Z.39.88. L'implementazione del *link resolver* di Van de Sompel fu acquistata da Ex-Libris e commercializzata col nome di SFX; cfr. Bucchioni [2002].

30 Cfr. Bucchioni – Spinelli [2007 p. 73-74].

31 Cfr. Tajoli [2005].

univoco di un qualsiasi oggetto di contenuto intellettuale a livello digitale. Il DOI nasce nella metà degli anni '90 per rispondere alla necessità, da parte degli editori, di identificare in maniera univoca un oggetto di proprietà intellettuale distribuito attraverso la rete. Diventato standard ANSI/NISO Z39.84-2000 il sistema DOI associa un insieme di metadati alla risorsa; questa è identificata da una stringa che viene sottoposta ad un apposito risolutore (basato sulla tecnologia Handle System) in grado di indirizzare verso le informazioni associate attraverso i metadati da chi ha registrato il DOI³². A garanzia del sistema è nato un apposito consorzio, l'International DOI Foundation, che si occupa dello sviluppo e della promozione dello standard³³.

Un altro protocollo particolarmente rilevante connesso ai metadati è OAI-PMH (Open Archives Initiative Protocol for Metadata Harvesting). Sviluppato all'interno dell'Open Archive Initiative³⁴, definisce un meccanismo che opera attraverso il protocollo HTTP per il recupero di metadati descrittivi (formattati in XML) da uno o più archivi. Il protocollo si basa sull'esistenza di "data provider", che contengono i documenti depositati e ne espongono i metadati relativi, e di "service provider", che usano i metadati dei data provider per offrire servizi a valore aggiunto [De Robbio 2002]. Complementare ad OAI-PMH, ma sviluppato parallelamente dalla Library of Congress, è il protocollo standard Search/Retrieve via URL (SRU), che cerca di fornire una soluzione alle diverse sintassi utilizzate via HTTP per sottoporre query di ricerca e per visualizzarne i risultati³⁵. Anche se entrambi i protocolli si occupano

32 L'assegnazione e il mantenimento di un DOI è affidata a una serie di agenzie a pagamento (la più famosa è CrossRef). Cfr. Vitiello [2004].

33 Cfr. Paskin [2009].

34 L'Open Archive Initiative (OAI), nasce nel 1999 (cfr. Lagoze - Van de Sompel [2000]), con l'obiettivo di sviluppare e promuovere standard per l'interoperabilità che favoriscano la disseminazione online di informazione di tipo accademico-scientifico (cfr. OAI [2006]). Il progetto si inserisce nell'ambito degli open archives, ovvero archivi (*repositories*) di e-prints accademici ad accesso gratuito, nati per contrastare un paradosso della divulgazione scientifica. Questo paradosso consiste nel fatto che spesso i ricercatori sono spinti a cedere i diritti dei loro lavori a editori commerciali che li rivendono poi attraverso le loro riviste, queste sono successivamente acquistate dalle biblioteche degli stessi enti che finanziano le ricerche, i quali pagano così due volte i risultati della ricerca, prima con lo stipendio dei ricercatori e poi con l'abbonamento alle riviste. Cfr. Ridi [2007 p. 102-103].

35 Le query sono formattate in CQL e trasportate attraverso il protocollo SOAP, una raccomandazione del W3C per lo scambio di informazioni strutturate. Il termine "SRU via HTTP SOAP" sostituisce la precedente dicitura "Search/Retrieve Web Service" (SRW). Cfr. SRU Editorial Board [2008].

di recupero di metadati da risorse remote, OAI-PMH si offre come alternativa di più facile implementazione e particolarmente orientata all'interoperabilità³⁶, mentre SRU, anche se più complesso, ha come obiettivo di riprodurre le funzionalità essenziali di Z39.50 nel campo della ricerca distribuita³⁷.

Per quanto riguarda il recupero di informazioni bibliografiche già in fase di acquisizione, lo sviluppo di standard per l'interoperabilità ha portato al confronto tra biblioteche ed editori sul tema dei metadati. È stata quindi promossa l'adozione di EDI (Electronic Data Interchange)³⁸ e delle sue implementazioni, come EDIFACT (EDI for Administration, Commerce and Transport), un insieme di standard che definisce norme e linee guida per lo scambio elettronico di dati strutturati, in particolare in relazione alle transazioni di beni o servizi³⁹.

Viene inoltre promossa la mappatura tra ONIX e MARC21 per permettere un più agile scambio di dati tra fornitori e biblioteche, essendo ONIX, uno standard basato su XML, ampiamente diffuso nel campo dell'editoria per la descrizione di libri, periodici e licenze di pubblicazione⁴⁰.

Per quanto riguarda la circolazione dei documenti, di recente si è affacciato al mondo delle biblioteche il sistema di identificazione a radiofrequenza (RFID, radio frequency identification), una tecnologia ampiamente usata nei sistemi antitaccheggio⁴¹. Si tratta di “un sistema di identificazione automatica in radiofrequenza basato su microchip inseriti in qualsiasi oggetto e capaci di inviare e ricevere segnali radio immediatamente convertiti, da appositi lettori, in dati digitali” [Di Giammarco 2004]. L'integrazione in questo caso consiste nella semplice implementazione dello scambio di dati tra i microchip e il sistema di automazione

36 Cfr. Hodgson [2002 p. 41].

37 Cfr. Morgan [2004a], Standerson [2005].

38 Electronic Data Interchange (EDI) fa riferimento allo scambio di dati strutturati in via elettronica tra due organizzazioni. Utilizzato prevalentemente nell'ambito dell'e-commerce, ha portato allo sviluppo di diversi standard (p.e. UN/EDIFACT, ANSI ASC X12, TRADACOMS). Cfr. WIKIPEDIA [2010b]

39 EDIFACT è standard ISO 9735. Per un'introduzione al tema in italiano cfr. Guerra – Pelizzari [2001].

40 Grazie al lavoro della Library of Congress sulla definizione XML del MARC21, sono state possibili anche altre mappature del MARC, tra cui quelle con: DublinCore, MODS, UNIMARC, Digital Geospatial Metadata. Cfr. Network Development and MARC Standards Office [2009].

41 Il sistema è già stato implementato all'estero presso le biblioteche pubbliche di New York, la Windsor Public Library e altre biblioteche universitarie. In Italia presso la Biblioteca Apostolica Vaticana. Cfr. Di Giammarco [2004].

bibliotecario, ad esempio, per la gestione dei prestiti, piuttosto che per controlli inventariali.

Sempre a proposito di circolazione, è stato introdotto, da parte dell'azienda americana 3M, lo Standard Interchange Protocol (SIP), che permette l'identificazione dell'utente e il conseguente self-check dei materiali. A questo standard de-facto ha fatto seguito lo standard NISO Z39.83 (NISO Circulation Interchange Protocol, NCIP), un protocollo di comunicazione nato appositamente per automatizzare le operazioni di prestito e favorire le applicazioni self-service e il prestito interbibliotecario⁴².

La pervasività del web ha portato le biblioteche all'inevitabile confronto con l'esplosione del fenomeno "web 2.0". Il termine, dalla definizione vaga⁴³, è utilizzato per indicare l'evoluzione in atto nello sviluppo dei servizi offerti attraverso la rete, che vedrebbero una maggiore centralità dell'utente, connessa ad un aumento dell'interoperabilità dei sistemi software permessa dall'ampia diffusione degli standard. Se da un punto di vista concettuale il termine è stato ripreso per definire la "Library 2.0"⁴⁴, provocando diversi dibattiti, è innegabile come i nuovi servizi e le nuove tecnologie che lo caratterizzano siano ampiamente sperimentate in ambito bibliotecario.

Per quanto riguarda la centralità dell'utente e la possibilità di personalizzazione dei servizi sono stati introdotti strumenti come: blog, wiki, feed RSS, folksonomies, social tagging, collaborative filtering e altri strumenti derivati dalle esperienze di comunità dei social network⁴⁵.

Dal punto di vista tecnologico, l'interattività è permessa dall'utilizzo di applicazioni

42 Le operazioni di comunicazione per il prestito interbibliotecario erano già state oggetto di standardizzazione nel 1997 con l'adozione della seconda edizione degli standard ISO 10160 e ISO 1016 che definiscono i termini e il protocollo di comunicazione utilizzato tra sistemi per lo scambio di documenti. Cfr. Neville – Riding [2004].

43 Il termine è stato coniato da Tim O'Reilly e John Battelle durante la "Web 2.0 Conference" tenuta a San Francisco nel 2004. Cfr. O'Reilly [2005]. La vaghezza consiste nel fatto che si tratta di una metafora (cfr. Van den Boomen [2007]) che vorrebbe racchiudere in sé tutte le ultime tendenze del web. Inoltre viene spesso usato con enfasi ma inappropriatamente, per la diffusione di prodotti o servizi commerciali.

44 Cfr. Casey [2005], Notess [2006].

45 Cfr. Ridi [2007], Lankes - Silverstein - Nicholson [2007], Marchitelli [2008b].

AJAX o DHTML⁴⁶, mentre per quanto riguarda lo sviluppo di applicativi web ci si basa sempre di più su piattaforme LAMP⁴⁷ e sulla diffusione delle API (Application Programming Interface), ovvero un insieme di procedure relative ad un'applicazione che le permettono di scambiare dati con le altre, facilitando così il lavoro del programmatore nell'integrazione di software diversi e dando origine ai cosiddetti web services e mash-up, applicazioni web create dall'ibridazione di informazioni o contenuti provenienti da diverse fonti.

La diffusione di queste applicazioni ha contribuito a concepire il web come vera e propria piattaforma per la diffusione di servizi fruiti direttamente online, il cui esempio principale sono i cosiddetti *Software as a Service* (SaaS), o come sono stati tradizionalmente chiamati *servizi in modalità ASP* (Application Service Provider)⁴⁸. Grazie infatti all'aumento dell'affidabilità e capacità dei servizi di hosting si è andata consolidando l'opportunità di offrire il mantenimento dell'applicativo e dei dati su server gestiti dal rivenditore, abbattendo così i costi legati all'acquisto e alla manutenzione di hardware e alla formazione locale di personale tecnico.

2.4. Verso nuovi paradigmi

L'integrazione dei numerosi strumenti a disposizione per l'accesso e la localizzazione delle risorse non è semplice e ha portato ad un confronto tra sito web e OPAC della biblioteca.

Il sito web tenderebbe a trasformarsi in portale⁴⁹, ovvero ad adottare

46 AJAX è l'acronimo di Asynchronous JavaScript and XML e indica una tecnica di sviluppo per la realizzazione di applicazioni web interattive (cfr. WIKIPEDIA [2010a]), DHTML sta per Dynamic HTML ed è un termine generico per fare riferimento ad un insieme di tecnologie che permettono di cambiare in modo dinamico la rappresentazione e il contenuto di un documento (cfr. WIKIPEDIA [2009]).

47 LAMP è un acronimo che indica le tecnologie coinvolte nella realizzazione dell'applicazione web: sistema operativo Linux, web server Apache, database MySQL, linguaggio PHP, Perl o Python. Esistono anche sistemi AMP basati su altri sistemi operativi: MS Windows (WAMP), Mac OS (MAMP), Solaris (SAMP), e OpenBSD (OAMP).

48 Cfr. Breeding [2009a], Bertini [2002].

49 Cfr. Ridi [2007 p.116], Pasqui [2006], Cassella [2007].

un'organizzazione dell'informazione orientata all'utente e in grado di offrirgli tutte le risorse utili (link al catalogo, meta-searcher, link resolver, indici di database, virtual reference desk⁵⁰, ecc.). L'OPAC dovrebbe invece rinnovarsi per "garantire l'accesso a un contenuto molto più ampio di quanto offerto tradizionalmente" e offrire "una funzionalità notevolmente potenziata, basata in prevalenza su caratteristiche standard dei motori di ricerca web e delle librerie online" [Byrum 2005]⁵¹.

Per questo da qualche tempo si fa riferimento agli OPAC di nuova generazione con Next Generation Catalog⁵², ovvero cataloghi in grado di realizzare al meglio il paradigma del Discovery to Delivery (D2D), secondo il quale il catalogo dovrebbe permettere di: scoprire l'esistenza della risorsa (*discover*), scoprire i servizi connessi a quella risorsa (*locate*), richiedere un servizio (*request*), fornire il servizio (*deliver*)⁵³. Anche per questo motivo, e in relazione alle tecnologie utilizzate per implementarli, sempre più di frequente si fa riferimento ai cataloghi con Discovery Layer Interface. Nell'era dell'integrazione, paradossalmente, proprio i sistemi integrati sembrano essere un problema da risolvere. Diventati una parte centrale dell'automazione bibliotecaria, per vari motivi⁵⁴ non sono ancora in grado di fornire un'adeguata risposta alle esigenze del mercato dell'informazione⁵⁵. Questi sistemi, che sono stati pensati soprattutto per la gestione della collezione fisica, non hanno ancora sviluppato pienamente le potenzialità di integrazione con la gestione dei vari tipi di risorse elettroniche, come:

- i materiali con licenza, come e-journal, servizi di Abstract&Indexing (A&I) e così via. Per questo ci sono strumenti come: ERM, metasearcher, resolver;
- i materiali digitalizzati dalla biblioteca;
- gli archivi istituzionali o gli archivi di learning object⁵⁶;

50 Con il termine "virtual reference desk" si intendono elenchi di risorse selezionate, possibilmente affiancati da servizi di reference online.

51 Oltre all'intervento di Byrum [2005], è stato provocatorio ma efficace anche quello di Schneider [2006] con il suo "How OPACs Suck"

52 Cfr. Morgan [2006].

53 Cfr. Dempsey [2005b].

54 Cfr. §3.4. La dis-integrazione dei sistemi integrati

55 Cfr. Dempsey [2005a].

56 Un learning object è una unità di istruzione per l'e-learning.

Oltre a questo, non vanno escluse le possibilità di integrazione con sistemi gestionali amministrativi o finanziari legati all'istituzione connessa alla biblioteca⁵⁷.

Per rispecchiare meglio questo nuovo flusso di lavoro della biblioteca è stato sviluppato l'Open Library Environment Project (OLE Project), che ha delineato un nuovo schema per il sistema automatizzato di prossima generazione, in grado di rappresentare la natura mutevole dei nuovi materiali gestiti e con l'obiettivo di portare le biblioteche ad assumere un ruolo centrale nel processo di ricerca⁵⁸. Il 20 ottobre 2009 è stato diffuso il rapporto finale della prima fase dell'OLE Project, che delinea le caratteristiche e i processi dell'OLE framework, ovvero di quell'insieme di applicativi che implementeranno le caratteristiche del progetto.

La tendenza generale che si può osservare è comunque legata ad una richiesta di maggiore apertura da parte dei sistemi integrati, che dovranno essere in grado di slegarsi dall'attuale modello su cui sono stati sviluppati e abbracciare nuovi paradigmi legati alla pervasività del web⁵⁹.

57 Esigenza sentita particolarmente soprattutto nel campo dell'alta formazione. Cfr. Courant – Griffiths [2006].

58 Cfr. OLE Project [2009].

59 Cfr. Dempsey [2009], Pasqui [2009].

3. Integrated Library System

3.1. Premessa

Con Integrated Library System (ILS)⁶⁰, si intende un applicativo sviluppato per gestire in forma elettronica l'insieme delle procedure che si svolgono in biblioteca.

Questo tipo di software pone l'accento sul fatto che i moduli sviluppati per automatizzare le diverse procedure (acquisizione, catalogazione, circolazione, periodici, OPAC) condividono lo stesso database e sono in grado quindi di creare un ambiente di lavoro omogeneo, sia per gli operatori, che possono agevolmente passare da un modulo ad un altro, sia per gli utenti della biblioteca, che possono accedere tramite un'unica interfaccia a funzioni di ricerca e gestionali [Weston 2007 p. 221]. In questo senso il termine *integrated* fa riferimento al fatto che i vari moduli condividono gli stessi dati in un unico database.

I sistemi integrati, che possono anche essere il risultato della connessione di moduli sviluppati da produttori diversi, hanno comportato diversi vantaggi nel campo dell'automazione: eliminando gli sforzi necessari a mantenere diverse basi di dati su cui replicare gli stessi dati; riducendo le possibilità di errore del record bibliografico, che viene registrato una sola volta; facilitando la gestione e la manutenzione del sistema.

⁶⁰ ILS non è l'unico termine con cui sono conosciuti i sistemi integrati, sono usati anche: Library Management Systems (LMS), Library Automation Systems (LAS), Integrated Library Management Systems (ILMS), Integrated Online Library Management Systems (IOLMS), Système Intégré de Gestion de Bibliothèque (SIGB), Sistema Integrado de Automatización de Bibliotecas (SIAB), Sistemi di gestione bibliotecaria, Sistemi di automazione per biblioteca.

3.2. Breve evoluzione degli ILS

I primi applicativi di questo genere nascono verso la fine degli anni '70, quando gli esperti di automazione di sistemi di circolazione iniziarono ad enfatizzare i vantaggi di avere un database comune di dati. Le prime soluzioni, sviluppate soprattutto dalle stesse biblioteche⁶¹, entrarono subito in competizione con i cosiddetti software per la circolazione "chiavi in mano"⁶², i quali ben presto iniziarono ad includere moduli aggiuntivi ai loro applicativi per poter rimanere competitivi⁶³. Fu il momento in cui nacque un vero e proprio mercato dell'automazione e le biblioteche passarono dallo sviluppo interno di applicativi all'acquisto di soluzioni commerciali in grado di rispondere alle loro esigenze.

A partire dagli anni '90, tuttavia, è iniziata una convergenza da parte dei produttori verso una serie di caratteristiche comuni ormai alla maggior parte dei sistemi integrati:

- adozione di un'interfaccia grafica per tutti i moduli funzionali;
- architettura client-server e protocollo di rete TCP-IP;
- database relazionali standard;
- supporto per UNICODE e sistemi multilingue;
- interfacce personalizzabili;
- OPAC consultabile dal web;
- conformità allo standard Z39.50;
- adozione del protocollo ILL (ISO 10160/61);
- compatibilità con lo standard EDI (Electronic Data Interchange);

⁶¹ Ad esempio l'University of Chicago Library fu una delle prime a implementare un sistema del genere, già alla fine degli anni '60, ma anche la Stanford University (che sviluppò BALLOTS), la Washington Library (WLN), la Northwestern University (NOTIS), e la Dortmunder Bibliotheks (DOBIS). Cfr Goldstein [1983].

⁶² Si trattava per lo più di sistemi per la gestione della circolazione, basati su mainframe o minicomputers, sviluppati in modo da essere adattabili a diverse realtà e venduti come pacchetto unico di hardware e software. Cfr Saffady [1989].

⁶³ Cfr. Saffady [1983].

- possibilità di avere contenuto accresciuto (ad es. integrazione con Amazon, Google Libri,...)

Le principali differenze che ancora permettono di distinguere diversi ILS si basano su: scalabilità, tipo di database utilizzato, compatibilità con diversi sistemi operativi, supporto per record MARC, interoperabilità con moduli aggiuntivi.

3.3. Principali funzionalità di un ILS

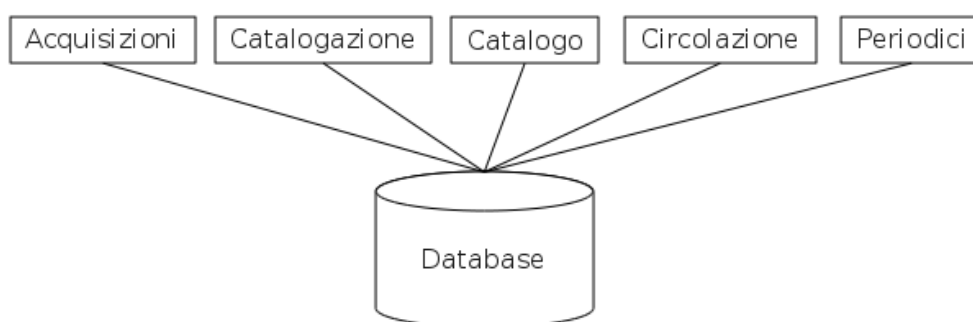


Figura 2: Moduli presenti in un sistema integrato

Attualmente ogni sistema integrato prevede un numero minimo di moduli in grado di gestire: acquisizione, catalogazione, catalogo online, circolazione, periodici. Ciascuno di questi moduli mette a disposizione una serie di funzionalità minime che ormai sono patrimonio comune della maggior parte degli ILS in circolazione.

3.3.1. Modulo per l'acquisizione

Il modulo per l'acquisizione dovrebbe prevedere come minimo:

- Elaborazione dei suggerimenti; permette di ricevere suggerimenti per l'acquisto da parte di utenti o staff e permette all'utente di tracciare lo stato

del suo suggerimento.

- Compilazione dell'ordine e aggiornamento; permette allo staff di elaborare e tenere traccia dell'oggetto da acquisire, permettendo l'importazione da un file esterno di proposte ed evitando le duplicazioni.
- Produzione dell'ordine e trasmissione; permette di trasmettere l'ordine al rivenditore (o editore, donatore, partner per lo scambio).
- Elaborazione dell'oggetto ricevuto; per l'aggiunta di eventuali dettagli e numero d'inventario.
- Elaborazione delle fatture ricevute.
- Reclami, permette di generare automaticamente reclami per oggetti in ritardo.
- Cancellazioni degli ordini.
- Preparazione delle etichette da applicare all'oggetto.
- Gestione finanziaria; permette la gestione dei fondi.
- Strumenti per la ricerca e il recupero di informazioni sullo stato delle acquisizioni.
- Gestione di report informativi.

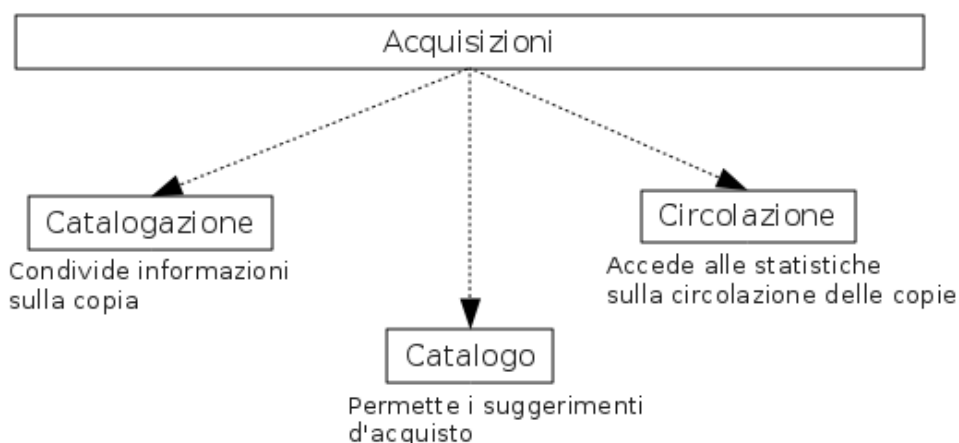


Figura 3: Esempi di comunicazione con il modulo acquisizioni

3.3.2. Modulo per la catalogazione

Il modulo per la catalogazione dovrebbe prevedere come minimo:

- Importazione ed esportazione di dati in formato standard⁶⁴, con possibilità di mappare il record importato secondo il database utilizzato dall'applicazione.
- Funzioni per la ricerca, il recupero e la visualizzazione dei dati, tipicamente implementate nell'OPAC.
- Creazione di record bibliografici; questi possono essere completamente nuovi, derivati da risorse esterne, creati utilizzando template diversi (p.e. basati su MARC), con possibilità di selezionare dati dall'authority file e di cambiare il tipo di lingua utilizzata⁶⁵.
- Verifica e validazione dei dati; ad esempio nel caso di una monografia si potrebbe controllare se sono stati compilati i campi obbligatori, oppure se non ci sono ripetizioni dei campi non ripetibili.
- Authority control, nel caso in cui la biblioteca mantenga i propri authority files.
- Stampa e produzione di output, come ad esempio etichette e codici a barre.

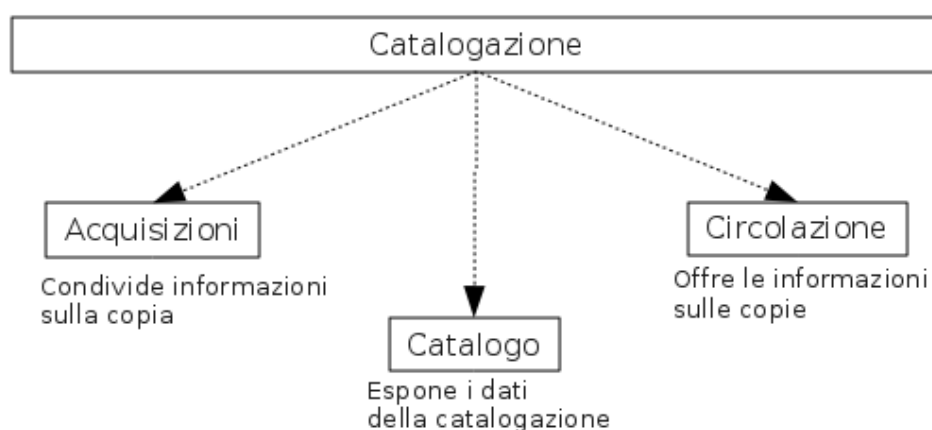


Figura 4: Esempi di comunicazione con il modulo catalogazione

⁶⁴ Si fa riferimento allo standard ISO-2709 (p.e. MARC21 e UNIMARC).

⁶⁵ A questo proposito è particolarmente utile il supporto di UTF-8 per la codifica dei caratteri.

3.3.3. Modulo per il catalogo online

Il modulo per l'OPAC dovrebbe prevedere come minimo:

- Accesso per autore, titolo, soggetto, ISBN, tipo di materiale, ecc.
- Ricerca booleana⁶⁶.
- Possibilità di navigare tra i record.
- Dettagli bibliografici, con possibilità di visualizzazione e scarico in diversi formati.
- Informazioni sulla disponibilità o per il prestito interbibliotecario.
- Area personale utente, con possibilità di salvare i record trovati.

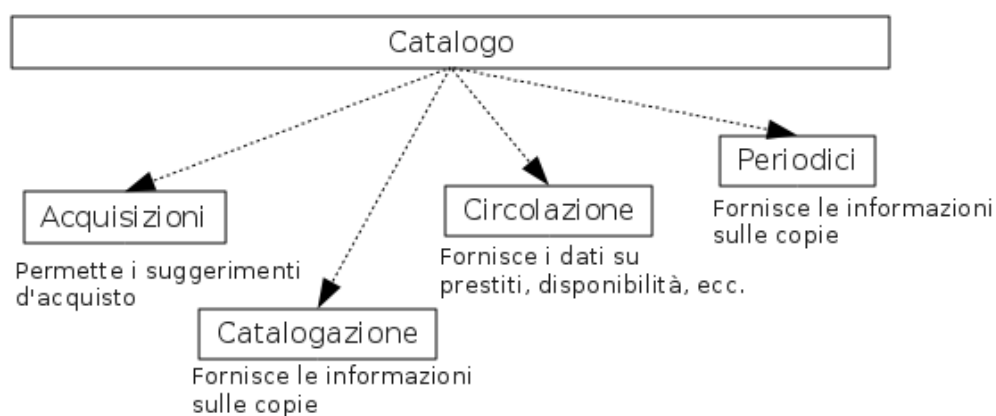


Figura 5: Esempi di comunicazione con il catalogo

3.3.4. Modulo per la circolazione

Il modulo per la gestione della circolazione dovrebbe prevedere come minimo:

- Registrazione degli utenti, con possibilità di gestione delle tipologie di membri da parte dello staff.
- Gestione degli addebiti, permette di gestire i prestiti fatti all'utente.

⁶⁶ La ricerca booleana permette di combinare tra loro i termini della ricerca attraverso gli operatori booleani AND, OR, NOT, raffinandone così il risultato.

- Prenotazioni del prestito, permette la gestione di una coda delle prenotazioni ed eventualmente di stabilire una tabella dei privilegi connessi alle tipologie di utenti.
- Rinnovo del prestito.
- Prestito interbibliotecario.
- Gestione dei ritardi, con possibilità di inviare automaticamente solleciti.
- Gestione delle multe.
- Gestione degli avvisi di rilegatura, nel caso in cui la biblioteca si appoggi a un servizio di rilegatura per la manutenzione dei materiali.
- Funzioni per la ricerca e il recupero di informazioni sulla circolazione da parte dello staff.
- Creazione ed elaborazione di rapporti informativi.

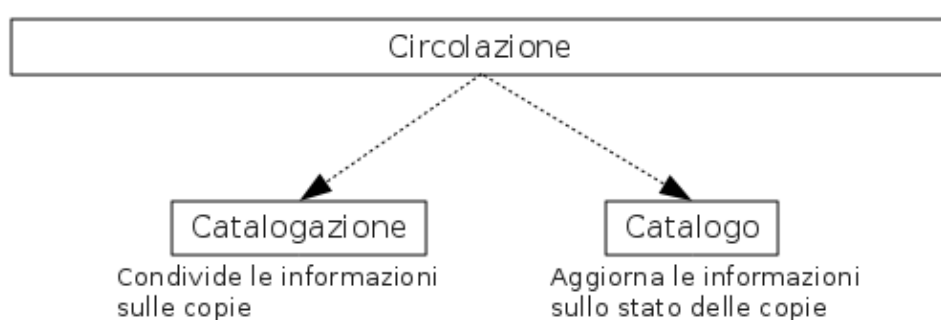


Figura 6: Esempi di comunicazione con il modulo circolazione

3.3.5. Modulo per la gestione dei periodici

Il modulo per la gestione dei periodici dovrebbe prevedere come minimo:

- Registrazione e gestione dei nuovi arrivi.
- Reclami, permette di generare automaticamente reclami per risorse in ritardo.
- Gestione degli avvisi di rilegatura, nel caso in cui la biblioteca si appoggi a un servizio di rilegatura per la manutenzione dei materiali.

- Lista unitaria dei seriali, nel caso di un sistema bibliotecario, dove più biblioteche cooperano assieme, permette di tenere sotto controllo i seriali.
- Gestione delle sottoscrizioni, permette di sottoscrivere abbonamenti, rinnovarli, gestire i fondi a disposizione.
- Funzioni per la ricerca e il recupero di informazioni sui seriali da parte dello staff.
- Preparazione delle etichette o codici a barre da applicare all'oggetto.
- Creazione ed elaborazione di rapporti informativi.

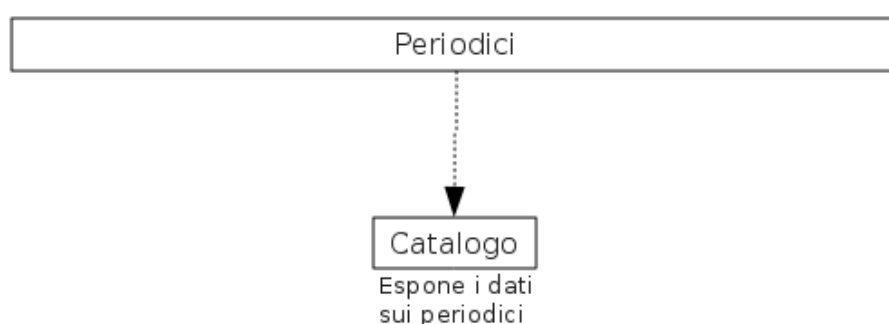


Figura 7: Esempi di comunicazione con il modulo periodici

3.4. La dis-integrazione dei sistemi integrati

Da quasi quindici anni, questo insieme di funzionalità, che rappresentano il cuore dell'ILS, ha iniziato ad essere affiancato da nuove funzioni⁶⁷, sia grazie alle possibilità derivanti dall'uso di standard per l'interoperabilità, sia di fronte alla concreta esigenza delle biblioteche ad affrontare il crescente mercato dell'editoria digitale.

Il problema principale per le biblioteche consiste nel fatto che queste nuove funzioni, invece di essere integrate nei moduli dell'ILS, sono offerte separatamente dal prodotto, causando un notevole aumento dei costi per l'automazione. I nuovi

⁶⁷ Strumenti per il metasearch o ricerche federate, Open URL link resolvers, RFID, ERMS, software per portali, cataloghi di nuova generazione, gestori di materiali digitali, software per archivi istituzionali, ecc. Cfr Pace [2004], Dempsey [2005a], Dougherty [2009].

moduli potrebbero comunque essere integrati nell'ILS mantenendo il sistema corrente e arricchendolo coi nuovi applicativi (da parte dello stesso rivenditore o con applicativi di rivenditori diversi), con il passaggio alla versione successiva del sistema in uso o, in alcuni casi, potrebbero forzare la biblioteca alla decisione di passare ad un nuovo sistema fornito da un diverso rivenditore⁶⁸.

Le cosiddette migrazioni, cioè i passaggi ad un nuovo software, sono raramente una decisione totalmente volontaria della biblioteca, ma sono spesso forzate dalla turbolenta situazione del mercato dell'automazione bibliotecaria. Pace [2005] ha definito questo mercato secondo 3 M: *markets, mergers, migrations*. “*Markets*” perché i principali rivenditori affermano spesso di poter operare in diversi settori dell'automazione, anche se in definitiva, per quanto riguarda le componenti principali dell'ILS, non si distinguono facilmente le implementazioni fatte per diverse tipologie di biblioteca. “*Mergers*” fa riferimento alle varie fusioni aziendali⁶⁹, che vedono una tendenza al consolidamento di alcuni grossi rivenditori grazie ad acquisizioni di realtà emergenti, e che possono essere, sul lungo termine, la causa della dismissione di sistemi integrati per cui viene interrotto lo sviluppo, limitandosi alla fornitura di supporto tecnico. “*Migrations*” indica il passaggio da un sistema integrato ad un altro da parte della biblioteca. Si stima ad esempio che il 63% delle vendite totali di ILS avvenute nel 2006 sia dovuto a migrazioni da sistemi ritenuti obsoleti⁷⁰.

In questo mercato, sembra che i produttori di ILS abbiano praticamente smesso di innovare i moduli base dei propri sistemi integrati, preferendo concentrarsi sullo sviluppo di componenti aggiuntivi, rivendibili separatamente⁷¹. Secondo alcuni bisognerebbe focalizzarsi piuttosto sullo sviluppo dell'OPAC, lasciando all'ILS esclusivamente le funzioni gestionali per l'acquisizione o la circolazione⁷².

68 Cfr. McGee [2007].

69 Pace [2005] fa riferimento in particolare all'acquisizione nel 2005, da parte di Sirsi Corporation, di Dynix, che ha creato SirsiDynix, un vero e proprio colosso nel campo dell'automazione.

70 Cfr. Breeding [2007a]. Breeding [2009b] individua in queste situazioni il tipico ciclo di vita di un ILS, che stima in una decina d'anni, prima che il “vecchio” (*legacy*) sistema integrato sia rimpiazzato dall'ultimo prodotto (*flagship*) del rivenditore.

71 Cfr. Cervone [2007], McGee [2007], Breeding [2009b],

72 Cfr. Tennant [2007], Balas [2007].

Anche se questo viene definito il periodo dell'interoperabilità tra software⁷³, da più parti viene richiesta una maggiore apertura dei sistemi integrati, attraverso la diffusione di API, l'accesso ai database, se non addirittura attraverso la diffusione del codice sorgente degli applicativi⁷⁴.

Una spinta verso una possibile ridefinizione dei sistemi integrati è rappresentata dall'ILS Discovery Interface Task Force, un gruppo di lavoro promosso nell'estate 2007 dalla Digital Library Federation (DLF) per cercare di risolvere le problematiche relative all'interoperabilità tra i tradizionali sistemi integrati e le applicazioni per la ricerca e scoperta di informazioni in rete. Nel marzo del 2008, il gruppo, formato dalle principali biblioteche accademiche americane, ha prodotto una serie di raccomandazioni sottoscritte poi da alcuni dei principali produttori di sistemi integrati e interfacce ("Berkeley Accord", 4 aprile 2008)⁷⁵. Queste raccomandazioni definiscono 4 livelli di interoperabilità che raggruppano al loro interno una serie di funzioni astratte che fanno riferimento ai servizi per la ricerca e fornitura di informazioni⁷⁶.

Nonostante tutto, sono state individuate due cause principali a quella che è stata chiamata dis-integrazione dei sistemi integrati⁷⁷: le procedure di acquisto di software da parte delle biblioteche e la sottostima del prezzo dei sistemi integrati da parte dei rivenditori.

Per quanto riguarda il primo punto viene considerato il modo in cui normalmente le biblioteche stilano i loro preventivi per i rivenditori, elencando le funzionalità aspettate dal nuovo software⁷⁸. Se da un lato questi preventivi hanno stimolato i

73 Cfr. § 2.3. Strumenti per l'integrazione.

74 Cfr. Bahr [2007], vedi anche § cap. 4. Software open source per l'automazione bibliotecaria.

75 I firmatari sono stati: Talis, Ex Libris, LibLime, BiblioCommons, SirsiDynix, Polaris Library Systems, VTLS, California Digital Library, OCLC, AquaBrowser. Cfr DLF ILS-DI [2008].

76 I 4 livelli sono strutturati in maniera gerarchica dal più basso al più alto, per cui il livello superiore include i livelli inferiori: 1-Basic discovery interfaces (BDI), 2-Elementary OPAC supplement, 3-Elementary OPAC alternative e 4-Robust/domain specific discovery platforms. Le funzioni sono raggruppate in: data aggregation (per l'estrazione e indicizzazione di dati dall'ILS), real time search (per l'accesso in tempo reale ai dati, viene raccomandato l'uso del nuovo protocollo SRU/W al posto di Z39.50), patron functionality (per l'interazione degli utenti con il sistema di circolazione/fornitura di documenti dell'ILS e per l'accesso alle informazioni personali nell'OPAC), OPAC interaction (per l'interazione con l'OPAC, sia da che verso strumenti di ricerca esterni). Cfr. DLF ILS-DI [2008], Pasqui [2009].

77 Cfr Dietz - Grant [2005], Breeding [2005].

78 Nello specifico si fa riferimento alla pratica, consolidata soprattutto nel mondo anglosassone, del

rivenditori ad adottare soluzioni standard e soddisfacenti, che potessero rispondere alle varie esigenze del mercato⁷⁹, dall'altro ha soffocato l'innovazione, contribuendo alla sostanziale immobilità nello sviluppo di software integrati⁸⁰.

Il secondo punto riguarda il fatto che i rivenditori, avendo già ampiamente recuperato i costi di creazione del software, si permettono di rivenderlo sul mercato ad un prezzo relativamente basso, appetibile anche alle realtà più piccole, contribuendo a una sottostima di quelli che potrebbero essere i costi dei servizi aggiuntivi.

3.5. Criteri di selezione di un sistema integrato

È utile notare che prima dell'acquisto la biblioteca dovrebbe considerare come, proprio a causa di questo fenomeno di dis-integrazione, i costi post-installazione rischiano di aumentare esponenzialmente. Spesso durante la fase di scelta conta di più il prezzo del prodotto “chiavi in mano” e si tendono a non considerare i costi nascosti⁸¹ dell'acquisto di un ILS, come ad esempio la formazione e l'assunzione di personale tecnicamente preparato, piuttosto che la pianificazione di adeguamento del sistema quando questo risulterà obsoleto.

A questo proposito è stato utile lo sviluppo del modello di Total Cost of Ownership (TCO) per le biblioteche, proposto da Lugg e Fischer [2003]. In questo modello, nella decisione dei fondi da stanziare per l'acquisto di sistemi automatizzati dovrebbero concorrere i costi relativi a:

- prezzo d'acquisto dell'applicativo, comprendente tutti quei moduli richiesti dalla biblioteca;
- servizi standard del rivenditore, solitamente legati alle operazioni di migrazione, testing e training, nonché agli accordi sulla manutenzione

Request for Proposal (RFP), attraverso il quale si richiede al rivenditore in che modo intende soddisfare certe richieste di funzioni e servizi.

79 Cfr. Hodgson [2002], Neville – Riding [2004].

80 Cfr. Breeding [2005], Cervone [2007].

81 Cfr. Barreau [2001]

dell'applicativo e sul servizio di assistenza;

- gestione operativa, ovvero tutti quei costi legati all'utilizzo dell'ILS nel tempo e che possono riguardare l'adattamento alle nuove esigenze degli utenti oppure l'aggiornamento del software;
- dismissione del sistema, ovvero i costi legati al cambiamento dell'applicativo derivanti dalla sua obsolescenza o sottoutilizzo.

Nonostante tutto non è sempre facile effettuare una comparazione dei prodotti in circolazione anche a causa della frammentazione del mercato, la cui analisi comporta la considerazione delle peculiarità nazionali⁸² o delle diverse tipologie di biblioteca (accademiche, pubbliche, specializzate, ecc.).

A questo va aggiunta la difficoltà a reperire dati quantitativi soddisfacenti sulla diffusione dei prodotti, per la cui raccolta ci si affida a indagini a campione⁸³ o rapporti annuali⁸⁴, se non addirittura a gruppi di studio interni alla biblioteca.

Proprio per questo motivo, ogni valutazione circa l'adozione di un sistema integrato, dovrebbe essere preceduta da alcune considerazioni sul rivenditore del prodotto⁸⁵. Intanto è necessario ottenere delle informazioni sulla sua stabilità finanziaria e se è inserito nel mercato anche con altri prodotti oltre al sistema integrato. In questo senso bisognerebbe porre particolarmente attenzione alle strategie di sviluppo del rivenditore, ovvero se investe ancora nello sviluppo dell'ILS o piuttosto concentra i suoi sforzi su altri prodotti.

Se si tratta di un rivenditore indipendente, anche se potrebbero esserci meno garanzie di stabilità finanziaria, dovrebbe essere più difficile un cambio di strategia di sviluppo. Nel caso invece di rivenditori legati ad aziende con più ampi interessi, bisognerebbe osservare se la strategia di sviluppo perseguita è inserita in un piano

82 È il caso dell'Italia, dove la connessione a SBN è un requisito spesso necessario e dove non sembrano avere grossa diffusione rivenditori molto presenti all'estero (p.e. SirsiDynix, Innovative Interfaces). Cfr. Bertini [2009].

83 Ad esempio, in riferimento al mercato delle biblioteche accademiche, cfr. Li [2009] per la Cina, Wang [2009] per gli Stati Uniti, Adamson - Bacsich - Chad - Kay - Plenderleith [2008] per l'Inghilterra. Su alcune criticità delle indagini a campione cfr. Scott [2010] e Corrado [2010] a proposito dell'indagine condotta in Breeding [2010] sul mercato internazionale di rivenditori di ILS.

84 Cfr. Bertini [2009] per l'Italia o Maisonneuve [2009] per la Francia.

85 Cfr. Cervone [2007]

strategico dell'azienda.

Non va sottovalutato, inoltre, che lo studio di fattibilità condotto dalla biblioteca durante la scelta del software, non deve concentrarsi esclusivamente sullo studio delle funzionalità offerte con il sistema integrato, ma dovrebbe analizzare in maniera preliminare l'insieme di procedure utilizzate all'interno del sistema informativo bibliotecario ed essere in grado di descrivere i flussi operativi esistenti e automatizzabili, avendo come obiettivo la definizione di un modello organizzativo in grado di integrarsi coi nuovi strumenti che verranno selezionati⁸⁶.

3.6. Ultime tendenze e possibili sviluppi futuri

Le attuali tendenze nel mercato degli ILS sono, dopotutto, le stesse che stanno coinvolgendo il mercato dell'information technology (IT) e che vedono lo sviluppo del cloud computing, termine per indicare l'utilizzo di hardware o software distribuito in remoto attraverso la rete. Questo tipo di approccio ha permesso la diffusione dei cosiddetti Software as a Service (SaaS), ovvero l'utilizzo di software attraverso internet, che ha permesso di abbattere i costi legati all'acquisto e alla manutenzione di hardware e alla formazione locale di personale tecnico⁸⁷.

I SaaS sono un tipico esempio dell'utilizzo del paradigma Service-Oriented Architecture (SOA), per cui sono messi a disposizione attraverso la rete diversi servizi posti sotto il controllo di domini/proprietari diversi. Questo paradigma permette lo sviluppo di applicazioni indipendenti dalle piattaforme, aderenti agli standard e soprattutto con una separazione tra il livello delle applicazioni e quello dei servizi forniti che ne aumenta le possibilità di interoperabilità⁸⁸.

In definitiva, è proprio un aumento dell'interoperabilità, soprattutto attraverso la diffusione di API, la richiesta più pressante fatta dalle biblioteche al mercato degli ILS. Interoperabilità che permetterebbe l'integrazione degli ILS anche con

⁸⁶ Cfr. Solimine – Weston – Fasella [1994].

⁸⁷ Cfr. Breeding [2009a], Breeding [2009b], § 2.3. Strumenti per l'integrazione.

⁸⁸ Cfr. Pasqui [2009], Breeding [2007b].

applicazioni non bibliotecarie, ERM, sistemi esterni di autenticazione e altri applicativi utili alla biblioteca per svolgere la sua missione. Anche per questo, in futuro, ILS potrebbe diventare l'acronimo per “interoperable library services”⁸⁹, ovvero un insieme di moduli funzionali in grado di dialogare tra loro.

⁸⁹ Cfr. Pace [2004] parlava di “interoperable library systems” e “integrated library services”. A mio avviso è ora possibile fondere i due termini, intanto perché *integrated* mi sembra più legato al risultato della connessione, mentre *interoperable* pone l'accento sulle potenzialità. *Services* per fare riferimento al paradigma SOA e al fatto che si tratta di vari moduli connessi piuttosto che un sistema monolitico.

4. Software open source per l'automazione bibliotecaria

4.1. L'approccio della "sorgente aperta"

Il termine open source, entrato da poco più di una decina d'anni anche nel lessico di ambienti non strettamente informatici, non è di facile disambiguazione, innanzitutto perché può assumere significati lievemente diversi rispetto al contesto di utilizzo, successivamente perché, avendo una storia relativamente giovane, è spesso vittima di facili pregiudizi.

In informatica open source indica un software per cui è possibile l'accesso al codice sorgente, ovvero rilasciato secondo determinate licenze che ne permettano lo studio e la modifica da parte di utenti terzi rispetto agli autori⁹⁰.

I software che siamo abituati a usare sono codificati in modo tale da poter essere capiti ed eseguiti dai computer, in sostanza sono composti di sequenze di zero e uno (linguaggio binario). Ovviamente i programmatori non scrivono in linguaggio binario ma utilizzano linguaggi di programmazione ad alto livello per scrivere tali programmi, quindi comprensibili da un essere umano. Il codice scritto con questo linguaggio è chiamato codice sorgente.

Usando una metafora⁹¹, immaginiamo per un momento di voler comprare una torta

⁹⁰ Cfr. INFONOMICS – BERLECON [2002], Riewe [2008].

⁹¹ Lievemente rielaborata da Colford [2009].

di compleanno per un nostro amico. Andando in pasticceria troviamo un vasto assortimento di torte diverse, purtroppo però su quelle con scritto Buon Compleanno c'è la frutta, che al nostro amico non piace, mentre le altre, alla cioccolata, sono decorate con la scritta Felice Anno Nuovo. Potremmo chiedere al pasticcere di fare una torta appositamente per noi, ma questo richiederebbe tempo e denaro in più.

Potremmo decidere invece di fare una torta per conto nostro, utilizzando come base la ricetta di un'amica e aggiungendo alcuni ingredienti che piacciono molto al nostro amico. Una volta preparata la torta potremmo anche scegliere di diffondere la ricetta per aiutare chi si trovasse in una situazione simile alla nostra.

In sostanza questo è l'approccio open source: abbiamo una ricetta (codice sorgente), fornita da qualcuno, a cui possiamo apportare delle modifiche e poi ridistribuire per aiutare altre persone.

Certamente non tutti sappiamo cucinare o potremmo non avere il tempo per farlo (ed è per questo che esistono diversi tipi di pasticcerie), tuttavia è evidente il vantaggio derivante dalla possibilità di avere accesso alla ricetta.

Proprio grazie ai molteplici vantaggi rappresentati dall'open source, sono nati nel tempo diversi movimenti, talvolta in contrasto sull'uso del termine, ma accomunati dall'obiettivo di diffondere tali vantaggi.

4.1.1. Free software, open source, FLOSS

La storia dell'open source va di pari passo con la storia dell'informatica. Fin dai primi anni '60 era comune tra gli sviluppatori di software condividere i codici sorgenti dei loro programmi, questo anche perché l'unica comunità interessata a tali programmi era la stessa comunità di sviluppatori. L'unico mercato per l'IT era quello della vendita e supporto per componenti hardware a un'utenza altamente specializzata. La tendenza alla condivisione si rafforzò ulteriormente con la nascita

delle prime reti, come Usenet, nata nel 1979 per connettere la comunità Unix⁹².

Dagli anni '80, con la nascita e rapida crescita dell'industria del software iniziarono a comparire anche i primi software a sorgente chiuso (o più comunemente *software proprietario*), rilasciati con licenze d'uso particolarmente restrittive⁹³.

Per ovviare a questo problema, nei primi anni'80 Richard Stallman, un programmatore del MIT, iniziò la sua attività per promuovere il software libero (*free software*), prima lanciando il progetto GNU⁹⁴, poi con la creazione della Free Software Foundation (FSF) nel 1985. Uno dei punti chiave per l'evoluzione del fenomeno open source fu la creazione della GNU General Public License (GPL), una licenza che concretizza le libertà fondamentali espresse da Stallman per descrivere il software libero:

- libertà di eseguire il programma per qualsiasi scopo
- libertà di studiare come funziona il programma e modificarlo per adattarlo alle esigenze dell'utente
- libertà di distribuire copie del programma per aiutare il prossimo
- libertà di migliorare il programma e condividere i miglioramenti fatti, di modo che la comunità possa trarne beneficio⁹⁵

Intorno alla FSF e alla filosofia del software libero si è sviluppato ben presto un vero e proprio movimento formato da utenti e sviluppatori di tale software, il cui prodotto forse più famoso è il sistema operativo GNU/Linux.

Anche se tale movimento si sviluppa ed è attivo per quasi vent'anni, il termine "open source" è relativamente giovane, essendo il risultato di una serie di incontri, avvenuti tra il 1997 e il 1998, tra un gruppo di esponenti del movimento in cerca di

92 Cfr. INFONOMICS – BERLECON [2002]

93 Uno dei casi più emblematici fu il cambiamento di politica della AT&T, distributore di Unix, che nei primi anni '80 iniziò a distribuire il sistema operativo con una licenza che ne restringeva l'utilizzo al solo acquirente, diversamente da quanto aveva fatto fino ad allora permettendo, ad esempio, alle strutture accademiche una licenza nominale per l'utilizzo su più macchine. I produttori di hardware (IBM, HP e DEC) iniziarono così a sviluppare sistemi operativi proprietari Unix imponendo ai loro programmatori un "accordo di non divulgazione" del software su cui lavoravano. Cfr. INFONOMICS - BERLECON [2002].

94 Acronimo ricorsivo per "GNU's not Unix".

95 Cfr. Bretthauer [2002], Colford [2009], Frigimelica - Marchitelli [2009].

un termine che potesse veicolare più efficacemente gli sforzi compiuti nello sviluppo di software al di fuori del modello proprietario⁹⁶.

A partire da questo gruppo, nel febbraio 1998, Bruce Perens e Eric S. Raymond, fondano l'Open Source Initiative (OSI), che propone un approccio più pragmatico alla questione del software open source e definisce le caratteristiche minime che devono avere la licenza e il software perché quest'ultimo possa definirsi tale:

1. Libera redistribuzione - la licenza non dovrebbe limitare in alcun modo la redistribuzione del software, anche se parte di un pacchetto software più grande, e non dovrebbe essere richiesto il pagamento di diritti per questa possibilità.
2. Codice sorgente - la distribuzione del programma deve prevedere il libero accesso al codice sorgente.
3. Prodotti derivati - la licenza deve permettere modifiche e prodotti derivati e questi devono poter essere distribuiti secondo gli stessi termini della licenza del software originale.
4. Integrità del codice sorgente dell'autore – la licenza può prevedere che le modifiche distribuite siano diffuse con un nome o numero di versione diversi dal software originale, o che le modifiche siano fatte tramite *patch files* piuttosto che attraverso la modifica del codice originale.
5. Nessuna discriminazione per persone o gruppi.
6. Nessuna discriminazione per ambiti specifici d'utilizzo – questo comprende, ad esempio, l'ambito commerciale.
7. Distribuzione della licenza – i diritti che accompagnano il programma devono essere applicati anche alle sue redistribuzioni.
8. La licenza non deve essere specifica ad un prodotto – un programma può essere estratto da una distribuzione più ampia ed essere usato e distribuito in accordo con i termini della licenza del programma.

⁹⁶ Cfr. Bretthauer [2002]. A conferma dell'importanza che il software libero raggiunse alla fine degli anni '90, basta citare il caso dei cosiddetti "Halloween Documents", una serie memorandum confidenziali prodotti all'interno di Microsoft e diffusi da Eric S. Raymond, dove viene segnalato come il software open source, e Linux in particolare, siano diventati possibili concorrenti di Microsoft.

9. La licenza non deve porre restrizioni ad altro software – se il programma è distribuito assieme ad altro software, la licenza non deve porre restrizioni su quest'ultimo.
10. La licenza deve essere neutrale rispetto alla tecnologia – non si possono porre restrizioni all'uso del programma su una determinata piattaforma o interfaccia.

L'OSI ha così approvato oltre 50 licenze che rispondono a questi 10 criteri, tra cui le più diffuse sono: Apache License, 2.0; new and simplified BSD licenses; GNU General Public License (GPL); GNU Library or "Lesser" General Public License (LGPL); MIT license; Mozilla Public License 1.1 (MPL); Common Development and Distribution License; Eclipse Public License⁹⁷.

Anche se Stallman preferisce usare il termine “free software”, piuttosto che “open source”, in quanto è fortemente convinto che il termine “free” veicoli meglio le quattro libertà fondamentali⁹⁸, si capisce facilmente che il principale motivo di dibattito tra la FSF e l'OSI ha radici più che altro filosofiche e i due movimenti si trovano in realtà a collaborare spesso.

Nella speranza di coniugare, almeno in ambito terminologico, entrambi i movimenti, sono state proposte alcune alternative, tra cui, di recente, la locuzione Free/Libre Open Source Software (FLOSS) per fare riferimento ai software rilasciati secondo licenze open source⁹⁹. Utilizzando anche la parola “libre” si pone l'accento sull'ambiguità, in inglese, del termine “free” per “free software”, spesso disambiguato dall'ormai celebre “free as in free speech, not as in free beer”, per cui non è una questione di prezzo ma di libertà.

Altro punto centrale nella definizione di “open source” riguarda il fatto che, oltre a indicare il modo di licenziare il software e il movimento per la promozione del suo utilizzo, il termine fa riferimento ad un certo modo di affrontarne lo sviluppo. Ci si trova di fronte a un modello di lavoro che incentiva la condivisione del sapere e il

⁹⁷ Per un elenco completo si veda il sito di riferimento dell'Open Source Initiative.

⁹⁸ Cfr. Stallman [2010b].

⁹⁹ Cfr. INFONOMICS – BERLECON [2002]. Un'altra variante utilizzata è Free and Open Source Software (FOSS o F/OSS).

supporto di una comunità che di volta in volta troverà il modo migliore per contribuire allo sviluppo¹⁰⁰.

Raymond [2000] usa la metafora della cattedrale e del bazaar per indicare i due modi principali di sviluppare software:

- Il modello della cattedrale si basa sulla centralizzazione della pianificazione ed esecuzione del lavoro ed è il modello caratteristico nello sviluppo del software proprietario. L'organizzazione del lavoro è verticale e fortemente gerarchizzata.
- Il modello del bazaar si basa sulla decentralizzazione della pianificazione ed esecuzione del lavoro e spiega bene il modello usato nello sviluppo di software open source. L'organizzazione del lavoro è reticolare, la comunità di sviluppatori lavora a parti di codice che poi condivide tramite patch e assembla assieme tramite Internet, usando un sito centrale di riferimento.

I due modelli proposti, comunque, non sono mutualmente esclusivi, anzi possono alternarsi durante lo sviluppo del software.

È stata proposta anche un'altra metafora che migliora quella del bazaar, ed è quella della “pentola”. Rispetto al bazaar, dove il valore dei beni scambiati non cambia durante i vari passaggi, in una pentola i vari ingredienti si combinano e vanno a formare qualcosa che ha un valore maggiore rispetto alla somma delle singole parti. Al di là delle metafore culinarie, forse l'analogia migliore per descrivere il modello di sviluppo open source è quella con l'atteggiamento assunto nella comunità accademica per quanto riguarda la ricerca scientifica¹⁰¹, dove l'accesso alle fonti, il peer-review e la divulgazione dei risultati sono fondamentali¹⁰².

100I contributi tipici sono, ad esempio: aggiunta/correzione di codice sorgente, creazione di documentazione, segnalazione o correzione di bug, traduzioni.

101Cfr. Sturman [2006].

102A questo proposito bisogna segnalare come di recente anche la comunità accademica si stia orientando verso un atteggiamento più open, grazie al movimento per l'Open Access che intende promuovere il libero accesso ai risultati della ricerca accademica. Cfr. Chudnov [2009], Di Donato [2009].

4.1.2. Open source tra mito e realtà

La rilevanza del fenomeno open source può essere percepita citando alcuni dei più importanti progetti in circolazione, spesso già ampiamente utilizzati inconsapevolmente:

- i sistemi operativi GNU/Linux e FreeBSD/OpenBSD;
- i linguaggi di programmazione Perl e Python;
- il linguaggio di scripting PHP;
- il web server Apache;
- i database MySQL e PostgreSQL;
- il software per la condivisione di file e stampanti Samba;
- il Berkeley Internet Name Daemon (BIND) che fornisce il Domain Name System (DNS), ovvero quel particolare sistema che traduce gli URL di un particolare sito in indirizzi IP¹⁰³;
- tutti i software basati sul sistema LAMP, ovvero la combinazione di Linux, Apache, MySQL e PHP. Gli esempi più famosi sono content management system (CMS) come Drupal, Joomla, Wordpress;
- la piattaforma per l'e-learning Moodle;
- la suite di strumenti per ufficio OpenOffice.org;
- il browser Firefox e il mail client Thunderbird;
- il programma per l'editing di immagini GIMP;
- il riproduttore multimediale VLC.

Anche se ampiamente diffuso, il software open source è ancora legato a una serie di pregiudizi, sia in senso positivo che negativo¹⁰⁴. Innanzitutto la credenza che il software sia gratuito.

Il software distribuito gratuitamente è chiamato *freeware* ed è a totale discrezione del distributore fornire anche il codice sorgente. In effetti esistono moltissimi software

¹⁰³Si potrebbe immaginare il sistema come una rubrica telefonica, per cui ci basta sapere il nome della persona da contattare e non il suo numero di telefono, che ci viene restituito automaticamente, appunto, dalla rubrica.

¹⁰⁴ Cfr. Potorti [2002].

proprietari distribuiti gratuitamente¹⁰⁵.

A questo proposito è stato spesso utilizzato il paragone “as free as a free kitten”, ovvero alla situazione in cui ci venisse regalato un simpatico gattino. Anche se non si paga niente all'inizio, dopo poco bisognerà iniziare a pagare per il cibo, per il veterinario, ecc.¹⁰⁶, ovvero per tutte quelle necessità, difficili da calcolare a priori, relative a personalizzazione, manutenzione, supporto, e che sono le stesse sia che si tratti di open source che di software proprietario.

Questo tipo di credenza è stato uno degli stimoli alla creazione della Open Source Initiative, poiché il termine “software libero” (*free software*) rischiava di trarre in inganno. In verità, quasi tutte le licenze open source (p.e. GPL) non impediscono di vendere il software¹⁰⁷, ponendo come unico vincolo l'accesso al codice sorgente, ma non menzionando i servizi che al software possono essere legati (p.e. documentazione, assistenza tecnica, hosting, garanzie, ecc.). Sono nate così altre forme di distribuzione del prodotto, non più legate alla vendita di singole copie, ma connesse ai servizi a valore aggiunto ad esso associati¹⁰⁸.

Un'altra credenza riguarda invece gli sviluppatori, che sarebbero tutti generosi volontari, quindi non in grado di offrire garanzie di stabilità al prodotto.

Anche se è effettivamente difficile quantificare la presenza di volontari nelle comunità di sviluppo, bisogna considerare che attorno al software libero sono nate diverse iniziative commerciali, legate soprattutto a interventi di installazione, manutenzione, personalizzazione o hosting. I dipendenti di queste aziende, come anche tutti gli utenti del software, anche non direttamente legati al campo dell'IT, sono incentivati a collaborare alle comunità di sviluppo del software per migliorarne le prestazioni¹⁰⁹.

¹⁰⁵ Un esempio per le biblioteche potrebbero essere software ampiamente utilizzati come MarEdit o CDS/ISIS.

¹⁰⁶ Cfr. Morgan [2004b].

¹⁰⁷ Cfr. Stallman [2010a] preferisce usare “distributing free software for a fee” al posto di “selling software”.

¹⁰⁸ Ad esempio RedHat, un'azienda che vende un sistema operativo basato su Linux assieme all'assistenza per il prodotto.

¹⁰⁹ A questo proposito è stata coniata da Raymond [2000] la cosiddetta Legge di Linus (facendo riferimento a Linus Torvalds, il “padre” di Linux), che dice “Given enough eyeballs, all bugs are shallow”, ovvero più persone sono coinvolte nel progetto meno sforzi saranno necessari a risolvere i problemi.

Altro pregiudizio riguarda invece il fatto che il software libero sarebbe privo di copyright. Anche se è possibile rinunciare al copyright su un programma, rendendolo così di pubblico dominio, la maggior parte del software libero è distribuito con una licenza, più o meno restrittiva in base alla tipologia¹¹⁰.

Un'ultima distinzione da fare è quella tra open source e open standard, cioè tutti quegli standard disponibili perché di pubblico dominio. Anche se è vero che gli open standard sono effettivamente studiabili e modificabili liberamente, trattandosi dell'unico modo per permettere l'interoperabilità tra i sistemi informatici esistono apposite istituzioni che si occupano di mantenerli stabili e diffonderli il più ampiamente possibile¹¹¹.

4.2. Open source e biblioteche

L'utilizzo di software open source in biblioteca può andare dall'adozione di applicativi più o meno generici, a programmi nati espressamente per l'ambito bibliotecario in grado di svolgere le varie funzioni necessarie ad un determinato servizio (p.e. ILS, digital repository, portali).

Per quanto riguarda il primo gruppo possono essere considerate varie soluzioni legate a prodotti come¹¹²:

- sistemi operativi basati su Linux, per le postazioni pubbliche o dello staff;
- lo strumento per il desktop publishing Scriblio o la suite di strumenti per ufficio OpenOffice.org;
- strumenti per l'istant messaging (IM) come Pidgin o Adium, utilizzati per il reference a distanza;
- il browser Firefox e le numerose estensioni attraverso le quali è possibile

110 Per alcune licenze, come la GNU GPL e la GNU LGPL, si è iniziato a parlare di "copyleft" in contrapposizione a copyright, per evidenziare il permesso dato dall'autore dell'opera al suo riutilizzo e diffusione.

111 Cfr. Chudnov [2009].

112 Cfr. Bisson [2007], Gordon - West [2008], West [2009].

personalizzarlo¹¹³;

- CMS basati o meno su piattaforme LAMP per la gestione del sito web della biblioteca;
- strumenti per il web-filtering come DansGuardian, SquidGuard o Untangle;
- convertitori di formato come PDFCreator;
- programmi per conferenze via web come DimDim;
- multidesktop manager come Groovix, per gestire le postazioni pubbliche;
- motori di ricerca come ht://Dig o Sphinx, integrabili nel sito web della biblioteca;
- sistemi per il *single sign-on*¹¹⁴ come Shibboleth;

Esistono poi diversi strumenti open source nati appositamente per l'utilizzo in biblioteca. È difficile quantificare il numero di progetti di questo tipo. Può trattarsi di applicazioni non più aggiornate, per mancanza di sviluppatori o perché considerate pienamente mature, oppure può trattarsi di soluzioni estremamente specifiche e poco diffuse¹¹⁵.

Attualmente l'attenzione maggiore viene riservata a OPAC e software per la gestione di collezioni digitali. Per quanto riguarda gli OPAC i progetti più noti sono:

- **Blacklight**

<<http://projectblacklight.org/>>

Licenza: Apache 2.0.

Sviluppato da: University of Virginia Library.

È attivo presso: Agriculture Network Information Center, Northwest Digital Archives, Stanford University, University of Virginia, University of Wisconsin-Madison.

- **Kochief**

<<http://code.google.com/p/kochief/>>

Licenza: GPL v. 3.

113 Cfr. Mercado [2007], Herzog [2008].

114 Traducibile con identificazione unica, permette di autenticarsi una sola volta e di accedere a tutte le risorse disponibili all'interno di un'organizzazione.

115 Solo SourceForge.net ne elenca un migliaio.

Sviluppato da: Casey Durfee insieme agli sviluppatori di Fac-Back-OPAC.

È attivo presso: Paul Smith's College Book Catalog, Drexel Libraries Collections.

- **Scriblio**

<<http://about.scriblio.net/>>

Licenza: GPL.

Sviluppato da: Casey Bisson, Lichen Rancourt, Jessamyn West, Cliff Pearson, per la Plymouth State University, supportato dalla Andrew W. Mellon Foundation.

Le installazioni riportate sul sito ufficiale sono presso: Lamson Library, Cook Memorial Library, l'archivio Beyond Brown Paper, Boston University School of Theology, Hong Kong University of Science and Technology.

- **Social OPAC (SOPAC)**

<<http://www.thesocialopac.net/>>

Licenza: GPL v.3.

Sviluppato da: John Blyberg, supportato dalla Darien Library.

Le installazioni riportate sul sito ufficiale sono presso: Darien Library, Palos Verdes Library District, Ann Arbor District Library.

- **VuFind**

<<http://vufind.org/>>

Licenza: GPL.

Sviluppato da: Villanova University's Falvey Memorial Library.

Sulla pagina ufficiale si contano circa 40 installazioni attive e il supporto per i seguenti ILS: SirsiDynix Horizon, Sirsi Unicorn, Voyager, VTLS Virtua, Innovative.

VuFind, Blacklight e Kochief si basano su Apache Solr¹¹⁶. VuFind è un software AMP (Apache, MySQL, PHP), Blacklight si basa su Ruby on Rails e Kochief su Django e

¹¹⁶Apache Solr è una potente piattaforma per la ricerca, basato sul progetto Apache Lucene, un motore di ricerca testuale basato su Java.

Python¹¹⁷.

Scriblio e SOPAC sono entrambi basati su due noti CMS open source di tipo LAMP. Il primo si basa su Wordpress , il secondo invece su Drupal ed è stato sviluppato assieme ai componenti Locum e Insurge¹¹⁸.

Un altro progetto interessante è eXtensible Catalog (XC), sponsorizzato dalla University of Rochester, dall'Andrew W. Mellon Foundation e altre istituzioni, che intende sviluppare una serie di applicazioni utili per l'accesso sia alla collezione digitale che fisica della biblioteca e pensato per integrarsi facilmente con content management systems o learning management systems. Iniziato nel 2007, ha sviluppato gli strumenti per l'integrazione con OAI e NCIP, attualmente sono in fase di sviluppo i componenti legati a: servizi con i metadati (normalizzazione, authority control, aggregazione, user-generated content, ecc.), integrazione con il CMS (Drupal), integrazione con il LMS (Blackboard).

I più noti software per la gestione di collezione digitali e la creazione di repository sono:

- **DSpace**

<<http://www.dspace.org/>>

Licenza: BSD.

Sviluppato da: MIT Libraries e Hewlett-Packard (HP).

Al momento, sul sito ufficiale del progetto, si contano quasi 900 repositories basati su Dspace.

- **EPrints**

<<http://www.eprints.org/>>

Licenza: GPL.

Sviluppato da: School of Electronics and Computer Science della University of Southampton.

Attualmente si contano circa 269 archivi basati su EPrints.

- **Fedora Repository Project**

¹¹⁷ Cfr. Bonfield [2009].

¹¹⁸ Si tratta di due componenti sviluppati sempre da Blyberg. Il primo permette di connettere l'ILS a SOPAC e fornisce un'interfaccia per il motore di ricerca full-text Sphinx. Il secondo permette di associare i dati "sociali" con i record bibliografici.

<<http://www.fedora-commons.org/>>

Licenza: Apache 2.0.

Sviluppato da: Sandy Payette e Carl Lagoze presso la Cornell University.

A gennaio 2010 erano elencati 172 progetti basati su Fedora.

- **Greenstone**

<<http://www.greenstone.org/>>

Licenza: GPL.

Sviluppato da: New Zealand Digital Library Project presso University of Waikato, inoltre è sviluppato e sostenuto in collaborazione con UNESCO e Human Info NGO.

Al momento, sul sito ufficiale del progetto, si contano quasi 70 repositories basati su Greenstone.

Nel luglio 2009 DSpace Foundation e Fedora Commons, le organizzazioni promotrici dei software citati, hanno iniziato ad operare insieme col nome DuraSpace. Un'altra azienda promotrice di soluzioni open source molto utilizzate soprattutto all'interno di sistemi integrati, è Index Data, che offre Zebra (un motore per l'indicizzazione basato su un database XML), Metaproxy e Pazpar2 (strumenti per metaricerche), i componenti YAZ (per lo sviluppo di applicativi Z39.50/SRU).

Oltre ai precedenti software esistono anche altri prodotti open source appositamente sviluppati per le biblioteche¹¹⁹: gestione seriali (CUFITS), metamotori di ricerca (LibraryFind), harvester OAI (MOAI, PKP OAI Harvester), gestionali per l'accesso alle postazioni pubbliche (Groovix, Pre-Book), content management systems (Library à la Carte, Kete), internet document delivery services (Prospero), questionari e sondaggi (Libstats, Instant Instruction Feedback Forms), gestionali del processo di digitalizzazione (DAF v2.0), strumenti per il reference (The Reference Portal).

119 Cfr. Library Success [2009].

4.2.1. ILS open source

Come nei casi visti precedentemente, anche per i sistemi integrati non è facile definire un numero esatto di progetti open source o stabilirne la diffusione.

I sistemi integrati attualmente più noti, per longevità o prestazioni, sono¹²⁰:

ABCD

<<http://bvsmodelo.bvsalud.org/php/level.php?lang=en&component=27&item=13>>

Sviluppato da BIREME a partire dal 2007 e supportato dal Flemish Interuniversity Council dal 2008, è completamente basato su software ISIS, un insieme di tecnologie sviluppate dall'UNESCO e distribuite gratuitamente.

Licenza: LGPL.

Tecnologie utilizzate: PHP, ISIS, Java, AJAX.

Moduli implementati: acquisizione, catalogazione, circolazione, OPAC, seriali.

Supporta: MARC, Z39.50, OAI-PMH.

Evergreen

<<http://www.open-ils.org/>>

Sviluppato a partire dal 2004 dal Georgia Public Library Service, è entrato in funzione nel 2006 presso le 252 biblioteche pubbliche della Georgia partecipanti al progetto PINES¹²¹. Il motivo principale che ha portato al suo sviluppo è stata l'inadeguatezza degli altri sistemi in circolazione per supportare un progetto come PINES. Ad oggi è utilizzato da più di 560 biblioteche.

Licenza: GPL.

Tecnologie utilizzate: è stato progettato secondo i paradigmi SOA e REST,

¹²⁰ Per le informazioni sui seguenti sistemi integrati, oltre ai siti ufficiali dei progetti, cfr. Sturman [2004], Boss [2008], Riewe [2008].

¹²¹ Il progetto PINES, avviato nel 1999, ha creato un sistema di circolazione nazionale, tramite la diffusione di una tessera unica per il prestito che permette di prendere a prestito e restituire materiali da una qualsiasi biblioteca partecipante al progetto.

utilizzando come linguaggi di programmazione Perl, C e Python per alcune componenti, il web server Apache, un database PostgreSQL e XUL¹²².

Moduli implementati: acquisizione, circolazione, catalogazione e OPAC. Nel 2010, con la nuova release, è previsto il rilascio del modulo per la gestione dei seriali.

Supporta: MARC21, SRU, Z39.50, SIP2, UNICODE, dalla prossima release EDI per il modulo acquisizioni.

Supporto commerciale fornito da: Equinox Software, Cherry Hill Company, ByWater Solutions, PTFS Europe, BibLibre.

Gnuteca

<<http://www.gnuteca.org.br/>>

Il progetto, sviluppato nel 2001 in Brasile come componente di un sistema integrato per l'università Univates di Porto Alegre, nel 2006 è stato valutato per l'adozione dall'Université Lyon II. Attualmente è utilizzato da diverse biblioteche scolastiche in Brasile.

Licenza: GPL.

Tecnologie utilizzate: Perl e PHP, Apache, PostgreSQL, MIOLO¹²³.

Moduli implementati: circolazione, catalogazione e OPAC.

Supporta: MARC21.

Supporto commerciale: Solis.

Koha¹²⁴

<<http://koha-community.org/>>

Sviluppato a partire dal 1999 in Nuova Zelanda dalla Katipo Communications Ltd. per la Horowhenua Library Trust, si è diffuso rapidamente anche negli Stati Uniti a

122 XML user interface language (XUL), è il linguaggio sviluppato dal Mozilla project e utilizzato dalle applicazioni di Firefox.

123 MIOLO è un framework per l'utilizzo di applicazioni in PHP, sviluppato e fornito da Solis, il distributore di Gnuteca, come sistema informativo integrato per le università

124 Cfr. § 6. Koha.

partire dal 2005. Attualmente è utilizzato da più di 900 biblioteche nel mondo.

Licenza: GPL.

Tecnologie utilizzate: Perl, Apache, MySQL. Utilizza Zebra per l'indicizzazione e la ricerca.

Moduli implementati: acquisizione, catalogazione, circolazione, OPAC, seriali.

Supporta: MARC21, UNIMARC, SRU, Z39.50, UnAPI, CoinS/OpenURL, OpenSearch, SIP2, OpenNCIP, UNICODE.

Supporto commerciale: Katipo Communications, LibLime, Equinox Software, Mill Run Technology Solutions, ByWater Solutions, PTFS, PTFS Europe, BibLibre, YourLibrarySite, Sabinet Online Ltd, Singapore Integrated Library Automation Services, Pakistan Library Automation Group (PakLAG), Libriotech, Catalyst, CILEA, Anant Corporation, Koha-India, LibSoul, OpenLX, OSS Labs, Tamil, Turo Technology LLP, inLibro, Calyx, Strategic Data, NCHC, Nucsoft OSS Labs.

LearningAccess ILS

<<http://www.learningaccess.org/tools/ils.php>>

Sviluppato da Griogair Bell viene ora distribuito gratuitamente dal Learning Access Institute alle piccole biblioteche pubbliche o scolastiche che non sono in grado di affrontare il costo d'acquisto di un sistema commerciale.

Licenza: GPL.

Moduli implementati: acquisizione, catalogazione, OPAC, gestione seriali, circolazione.

Supporta: MARC21, Z39.50.

NewGenLib

<<http://www.verussolutions.biz/web/>>

Sviluppato nel 2005 in India dalla Verus Solutions Pvt Ltd e dal Kesavian Institute of Information and Knowledge Management, è diventato un progetto open source nel

2008. Attualmente è installato in circa 120 biblioteche in Asia.

Licenza: GPL.

Tecnologie utilizzate: Java, PostgreSQL, Apache.

Moduli implementati: acquisizione, catalogazione, circolazione, OPAC, seriali.

Supporta: MARC21, UNICODE, SRU, Z39.50, OAI-PMH, RFID.

Supporto commerciale: Verus Solutions Pvt. Ltd.

OPALS

<<http://www.opals-na.org/>>

Sviluppato nel 2002 da 6 sistemi bibliotecari scolastici di New York, formato da circa 300 istituti, attualmente è utilizzato anche in un'altra dozzina di piccole biblioteche americane.

Licenza: GPL.

Tecnologie utilizzate: Perl, Apache, MySQL, Zebra.

Moduli implementati: OPAC, catalogazione, circolazione, ILL.

Supporta: MARC21, Z39.50, UNICODE.

Supporto commerciale: MediaFlex.

OpenBiblio

<<http://obiblio.sourceforge.net/>>

Sviluppato nel 2006, sembra essere stato adottato da una dozzina di piccole biblioteche nel mondo. L'attività di sviluppo della versione 1.0 è rallentata dal ridotto numero di persone che partecipa al progetto.

Licenza: GPL.

Tecnologie utilizzate: PHP, Apache, MySQL.

Moduli implementati: OPAC, circolazione, catalogazione.

Supporta: UNIMARC, Z39.50.

Non è previsto supporto commerciale da alcuna azienda.

PMB

<<http://www.sigb.net>>

Sviluppato nel 2002 da una biblioteca pubblica francese è attualmente sviluppato dall'azienda francese PMB Services. È molto diffuso soprattutto in Francia, con quasi 500 installazioni.

Licenza: CECILL (conforme a GNU/GPL).

Tecnologie utilizzate: PHP, Apache, MySQL.

Moduli implementati: acquisizione, catalogazione, circolazione, OPAC, seriali.

Supporta: UNIMARC, Z39.50, OAI-PMH, RFID.

Supporto commerciale: PMB Services.

Altri sistemi integrati open source conosciuti sono: phpMyLibrary, Avanti MicroLCS, Emilda e Pythea. I primi due sono portati avanti da un'unica persona, quindi estremamente lenti nello sviluppo e, forse, addirittura mai installati in nessuna biblioteca. Emilda è attivo in quattordici biblioteche scolastiche finlandesi con i moduli per la circolazione, catalogazione e OPAC; è compatibile con MARC e Z39.50, inoltre usa il software open source Zebra della IndexData per l'indicizzazione e la ricerca dei record. Purtroppo lo sviluppo è fermo al 2005 e l'azienda che lo supportava sembra essere sparita. Pythea è stato un progetto interno alla University of Windsor's Leddy Library e non più sviluppato dal 2002.

Si può notare facilmente, soprattutto da questi ultimi quattro progetti ma anche guardando a OpenBiblio, l'importanza di un'ampia comunità di supporto al software, sia di sviluppatori che di utenti. In questo senso, al momento i progetti più diffusi e con uno sviluppo costante, sembrano essere Evergreen e Koha.

4.2.2. Luci e ombre dei sistemi integrati open source

I potenziali vantaggi derivanti dall'utilizzo di ILS open source sono¹²⁵:

- la possibilità di personalizzare il prodotto per adattarlo alle proprie specifiche esigenze;
- l'assenza di restrizioni sull'utilizzo legate alle licenze;
- la possibile diminuzione dei costi; questo è sicuramente vero per quanto riguarda l'acquisto, ma vanno sempre considerati i possibili costi di sviluppo e manutenzione;
- una maggiore portabilità, grazie all'aderenza a numerosi standard;
- il modello open source, dove la comunità è abbastanza ampia, permette di aumentare le revisioni e i miglioramenti al codice;
- la flessibilità nella scelta di un fornitore di servizi, che potrà essere diverso dal rivenditore del sistema integrato.

I potenziali svantaggi invece riguardano:

- la difficoltà di prevedere i costi; spesso un sistema integrato open source non implementa efficacemente i moduli per l'acquisizione, la gestione dei seriali o l'ILL. Mentre potrebbero essere completamente assenti altri tipi di servizi (p.e. supporto per RFID, OAI harvester, self-check, ecc.). Implementare tali moduli per adattare il software alle esigenze della biblioteca potrebbe essere difficilmente calcolabile a priori. Vanno considerati anche possibili costi aggiuntivi legati al training del personale o alla produzione di documentazione (soprattutto quella per l'utente). Per questo è particolarmente utile il modello TCO e un approfondito benchmarking prima della scelta del prodotto;
- lo sviluppo decentralizzato del software, se da un lato offre numerosi vantaggi, in alcuni casi rende impossibile prevedere i tempi di release o di risoluzione di bug;

125 Cfr. Boss [2008], Riewe [2008]

- potrebbero mancare aziende specializzate in quel software, in grado di fornire training, documentazione e supporto tecnico;
- troppi pochi partecipanti allo sviluppo del progetto potrebbero causarne l'abbandono;
- scalabilità e velocità. Non tutti i sistemi integrati open source risultano adatti a realtà di grandi dimensioni.

A proposito di svantaggi, intorno al 9 settembre 2009 è stato diffuso tra un gruppo di clienti della SirsiDyNix, un testo chiamato "Integrated Library System Platforms on Open Source", prodotto da Abram [2009], vice presidente dell'innovazione alla SirsiDyNix. Il testo, che riporta il punto di vista molto critico dell'azienda sul software open source, ha sollevato parecchio scalpore¹²⁶ e da più parti è stato considerato una strategia FUD¹²⁷.

In effetti l'unica questione rilevante sollevata dal testo sembra essere il dubbio che lo sviluppo di ILS open source costituisca uno sforzo ripetitivo e non particolarmente utile in un panorama in cui i sistemi sembrano maturi e le priorità sono altre¹²⁸. Tuttavia questo non fa che confermare il problema relativo alla dis-integrazione dei sistemi integrati causata dai sistemi commerciali¹²⁹ e si inserisce nella questione più ampia circa il futuro degli ILS¹³⁰.

4.2.3. Criteri di valutazione di un ILS open source

Data la difficoltà di valutazione di un progetto open source affidabile, sono stati proposti alcuni criteri per aiutare la selezione di software di questo tipo¹³¹:

126 Cfr. Code4Lib [2009].

127 "Fear, Uncertainty and Doubt" (FUD) è una strategia che prevede la diffusione di informazioni negative e vaghe in grado di sollevare dubbi e incertezze su un concorrente, per scoraggiarne la scelta da parte di un acquirente. Il termine, nato proprio in ambito informatico già negli anni '60 per fare riferimento alla strategia di IBM nei confronti dei piccoli rivenditori, negli anni '90 è stato usato per indicare la strategia di Microsoft nei confronti di Linux. Cfr. Wikipedia [2010c].

128 Cfr. Breeding [2009c].

129 Cfr. Corrado [2009].

130 Cfr. §3.6. Ultime tendenze e possibili sviluppi futuri.

131 Cfr. Cervone [2003], Boss [2008].

1. Innanzitutto bisogna capire se il sistema continua ad essere sviluppato e, nel caso, se ci sono almeno due sostanziali release all'anno; inoltre si tratta anche di capire se e come è organizzata la comunità (blog, wiki, liste di discussione, ecc).
2. Il sistema integrato dovrebbe fornire come minimo i moduli per la catalogazione, la circolazione e l'OPAC; inoltre il modulo per le acquisizioni e per la gestione dei seriali dovrebbero essere in sviluppo.
3. Controllare se è previsto il supporto di formati MARC.
4. Controllare se è possibile ottenere il codice sorgente della versione corrente e la documentazione esistente.
5. Verificare se il prodotto è attualmente utilizzato in un numero significativo di biblioteche.
6. La scalabilità non dovrebbe essere un problema, ovvero non ci deve essere il rischio che l'ampiezza del database o la quantità di attività superino le capacità del software.
7. Nel caso fosse necessario sviluppare delle componenti mancanti bisognerà capire in che linguaggio di programmazione è scritto il programma e se nello staff sono presenti persone che lo conoscono (o se è possibile accedere a risorse esterne). Assicurarsi inoltre che il tipo di modificazioni possano essere integrate nel prodotto, per evitare di dover ripetere l'operazione ad ogni release.
8. Capire se si possiede già hardware e software sufficienti a far funzionare l'ILS o quanto costerebbe procurarli.
9. Capire se esiste e di che tipo è il supporto commerciale per quel prodotto.

Ovviamente, queste specifiche valutazioni andranno integrate con le altre, compiute comunemente prima della scelta di passaggio ad un nuovo software¹³².

132 Cfr. § 3.5. Criteri di selezione di un sistema integrato.

5. La situazione in Italia

5.1. La lunga via della cooperazione

Le esperienze di automazione bibliotecaria in Italia iniziarono presto, già con le schede perforate, ciononostante è possibile registrare un ritardo di qualche anno rispetto al percorso che ha caratterizzato le realtà anglosassoni¹³³. Oltre alle soluzioni nate internamente alle biblioteche fin dagli anni '60, si può dire che il ruolo centrale nel processo di automazione italiano l'abbia svolto il Servizio Bibliotecario Nazionale (SBN).

E' il 1979 quando si inizia a concretizzare il progetto di un "sistema bibliotecario nazionale". La prima conferenza nazionale delle biblioteche italiane, tenuta a fine gennaio, ruota infatti attorno alla necessità di creare un'infrastruttura nazionale per l'accesso all'informazione in grado di stimolare la cooperazione e la partecipazione da parte di tutti gli attori istituzionali¹³⁴. L'idea nasceva sulla scorta di esperienze di cooperazione già avviate negli anni '70 a livello locale¹³⁵ e con l'intento di armonizzare un panorama reso ancor più variegato dalla nascita delle regioni.

La partecipazione di queste nuove entità amministrative aveva notevoli implicazioni

¹³³ Cfr. §2.2. Breve storia dell'automazione bibliotecaria.

¹³⁴ Cfr. Leombroni [2002, p. 372].

¹³⁵ Si fa riferimento soprattutto alle esperienze dei sistemi bibliotecari provinciali toscani (Arezzo, Massa Carrara, Livorno, Firenze, Pistoia) a cui aderivano 80 enti locali (cfr. Leombroni [2002, p. 376]) e dell'Istituto Universitario Europeo, che, fin dalla sua apertura nel 1976, fu particolarmente attento al tema dell'automazione bibliotecaria e della cooperazione.

politiche: innanzitutto si sentiva la necessità di un dibattito più ampio rispetto alla possibile attività “centralizzante” del solo Ministero; in secondo luogo vi era la convinzione di dover definire in modo razionale le misure da prendere a livello locale e nazionale; non da ultimo la possibilità di affrontare meglio le problematiche relative al rapporto tra regioni e comuni in tema di biblioteche.

Sulla scia dei risultati della conferenza e di altri progetti di quegli ultimi anni¹³⁶, venne istituita con decreto ministeriale del 4 aprile 1980 una Commissione nazionale per l'automazione delle biblioteche. Questa produsse un documento che delineò due obiettivi principali: la creazione di un archivio bibliografico nazionale e l'organizzazione di una rete per la distribuzione delle risorse tra le biblioteche¹³⁷.

Concretamente questi obiettivi dovevano attuarsi con la creazione di una struttura centrale in grado di controllare la produzione della bibliografia nazionale e una struttura decentrata costituita dai servizi locali per l'accesso ai documenti.

Anche grazie ad un seminario tenutosi a Fiesole a cui parteciparono, oltre all'Istituto Centrale per il Catalogo Unico (ICCU), le Università di Bologna, Pavia e Firenze, la regione Toscana, la Bibliografia Nazionale Italiana (BNI) e l'Istituto Universitario Europeo, nel 1981 venne ulteriormente raffinata la visione del progetto e il “sistema” divenne “servizio”, usando un termine più aderente al modello decentrato e cooperativo che vedeva, accanto al ruolo centrale dell'archivio bibliografico nazionale, una determinante apertura ai servizi locali per la circolazione dei documenti.

Molto concretamente, gli obiettivi del Servizio erano:

- garantire la rapida localizzazione e circolazione dei documenti, tramite la creazione di un catalogo comune formato direttamente dalle biblioteche partecipanti al progetto;
- raggiungere un'ottimizzazione del lavoro dei bibliotecari evitando lavori

¹³⁶In particolare il programma Eco (Esperienza di cooperazione) e il conseguente progetto SNADOC (Servizio nazionale di accesso ai documenti) che, avviato nel novembre del 1979, prevedeva nella sua prima fase un servizio di circolazione tra le biblioteche toscane ma che intendeva articolarsi su quattro livelli (locale, regionale, nazionale e internazionale) per permettere, oltre al servizio di circolazione, anche un programma di catalogazione e le relative politiche di controllo bibliografico. Cfr. Leombroni [2002, p. 380-381]

¹³⁷ Cfr. Leombroni [2002, p. 382]

ripetitivi soprattutto a livello di catalogazione;

- impiegare in maniera ottimale le risorse destinate all'acquisto e conservazione dei documenti¹³⁸.

Ma il progetto era, in definitiva, ben più ambizioso. “Si può dire [...] che non si tratta di un progetto di automazione (che quindi sul piano dell'automazione vada affrontato), ma di un progetto che si serve dell'automazione per ottenere un nuovo disegno complessivo della realtà bibliotecaria italiana” Crocetti [1986].

Prima della metà degli anni '80 il progetto di automazione del “Servizio Bibliotecario Nazionale” (SBN) prevedeva la creazione di Poli, ovvero raggruppamenti locali di biblioteche, collegate in rete tra loro e a una macchina centrale (Indice) con funzioni di routing. Durante l'operazione di catalogazione sarebbe stata inviata una richiesta alla base dati locale e, in caso di esito negativo, all'Indice collegato alla rete nazionale, che avrebbe segnalato la presenza o meno della medesima registrazione bibliografica. In questo senso l'idea di catalogo unico era solo una visione logica unitaria di una struttura fisica decentrata che lasciava le basi dati in mano alle biblioteche locali¹³⁹.

Nel 1985, con la redazione di uno studio di fattibilità della rete nazionale affidato alla società italiana Italsiel e alla canadese GEAC, la prospettiva mutò. La società canadese valutò positivamente il progetto originario, compresa la scelta del modello OSI¹⁴⁰ per la realizzazione della rete; le uniche due raccomandazioni riguardavano un aumento degli oneri affidati all'Indice per la sua funzione di instradamento.

Lo studio parallelo condotto dall'Italsiel propose una rete di tipo stellare, per la comunicazione tra Indice e poli, e l'adozione, invece del modello OSI, di un protocollo proprietario di rete SNA, che avrebbe ridotto notevolmente i tempi di realizzazione della rete in quanto già impiegato sugli elaboratori IBM in dotazione.

Le discussioni sul modello tecnologico di riferimento furono accese e ruotarono soprattutto intorno al tema dell'interoperabilità permessa dall'adozione di protocolli

138 Cfr. Peruginelli – Giordano – Pettenati [1989, p. 19].

139 Cfr. Foglieni [1984].

140 L'Open Systems Interconnection (OSI) è uno standard di riferimento proposto dall'ISO (ISO 7498) che definisce un modello di riferimento per l'interconnessione di sistemi aperti.

standard (OSI). Ciononostante, alla fine prevalse la soluzione dell'Italsiel, molto probabilmente perché avrebbe permesso la presentazione in tempi brevi del progetto, in grado così di accedere ai generosi finanziamenti statali che proprio in quegli anni alimentavano il settore culturale¹⁴¹.

Di fatto il primo prototipo dell'Indice venne presentato solo nel 1990 e l'attivazione del primo collegamento tra quest'ultimo e le due Biblioteche Nazionali Centrali avvenne solamente nel 1992.

La realizzazione finale presentava un sistema fortemente centralizzato e oneroso, inoltre, di fatto, il progetto frenò la crescita di un mercato italiano dell'automazione bibliotecaria. Infatti, da un lato molte biblioteche pubbliche e sistemi bibliotecari avrebbero preferito un sistema più flessibile e meno costoso¹⁴², dall'altro si scoraggiò la creazione di applicativi non compatibili con SBN.

In definitiva, tra la metà degli anni '80 e i primi anni '90, il progetto SBN generò alcune dicotomie: tra sistemi proprietari e sistemi aperti, tra grandi biblioteche e piccole biblioteche, tra sistemi SBN e sistemi non SBN. Non da ultimo il fatto che l'intera infrastruttura tecnologica, nata secondo uno spirito di innovazione, rischiava di risultare fin da subito obsoleta di fronte alla nascita di Internet e alla diffusione dell'architettura client-server. Nonostante tutto, era iniziata la connessione all'Indice e, a partire dal gennaio 1993, la “migrazione” dei poli verso il sistema SBN¹⁴³.

Nel 1994 entrarono ufficialmente nel progetto anche le università, grazie ad un nuovo protocollo d'intesa tra Ministero per i Beni Culturali e ambientali, Ministero dell'Università e della ricerca scientifica e Presidente del Coordinamento delle regioni, che, oltre a formalizzare la partecipazione del mondo universitario a SBN, ne fissava una nuova struttura organizzativa.

Come primo passo verso un adeguamento tecnologico venne proposto “il progetto

141 Si fa riferimento agli stanziamenti, inseriti nella finanziaria del 1986 (Legge 28 febbraio 1986, n. 41), per lo sfruttamento dei cosiddetti “giacimenti culturali”, termine coniato dall'allora ministro del Lavoro Gianni De Michelis, convinto che fosse necessario investire nel patrimonio culturale italiano per aumentare i posti di lavoro. Cfr. Leombroni [2002 p. 402].

142 È il caso, ad esempio, delle biblioteche della provincia di Modena, che, a seguito di attenti studi di fattibilità (cfr. Pettenati [1985 p. 37-61]), scelsero il sistema Tinlib (cfr. Danesi – Giaccari [1986 p. 37-42]).

143 Non senza qualche difficoltà, derivante soprattutto dal tasso di duplicazione dei dati bibliografici sul sistema Indice. Cfr. Leombroni [2002, pag. 415].

Unix client-server” per la realizzazione di un applicativo per piattaforme Unix, che implicava l'abbandono di architetture proprietarie e l'adozione di una struttura più modulare¹⁴⁴.

L'applicativo ha introdotto un terzo livello locale (Biblioserver), oltre ai livelli Polo e Indice, per la distribuzione di dati e funzioni. In questo modo è stata aumentata l'autonomia a livello locale delle singole biblioteche che hanno accesso a una serie di funzionalità anche in assenza di collegamento al Polo.

Per quanto riguarda la distribuzione dei dati sui tre livelli vale il principio base su cui si fonda SBN, per cui si ha quasi sempre una replica sul livello superiore di un sottoinsieme dei dati contenuti solo a livello inferiore. In questo modo sul Biblioserver risiede un numero di informazioni, relative alla base dati locale, maggiore rispetto a quelle presenti sul Polo, mentre su quest'ultimo vi saranno meno informazioni circa la tipologia dei dati, ma comunque una maggiore quantità di record, poiché li riceve da tutte le basi dati locali che fanno riferimento a quel Polo.

Per quanto riguarda le funzioni sviluppate, è stata considerata necessaria la connessione al Polo (ed eventualmente all'Indice) solo per le informazioni condivise che richiedono allineamenti¹⁴⁵, mentre per le restanti funzioni su dati non condivisi non è richiesto il collegamento, eventualmente effettuato tramite operazioni batch¹⁴⁶.

Sono state sviluppate funzioni per:

- acquisizioni e gestione del documento fisico;
- gestione bibliografica, che consente la catalogazione sulla base dati di Polo degli spogli;
- gestione periodici;
- catalogazione semantica;
- gestione dei partner, intesi come biblioteche, bibliotecari, fornitori e utenti;

144 Cfr. ICCU [2010].

145 Con “allineamento” si intende la procedura attraverso la quale i dati presenti sulla base dati locale vengono aggiornati sulla base di correzioni effettuate da altri Poli/Biblioteche e distribuite tramite l'Indice. In questo senso l'allineamento assicura che l'informazione bibliografica sia uguale su tutti i cataloghi che la gestiscono.

146 Con “operazione batch” si indica, in informatica, l'esecuzione di più comandi o programmi raggruppati insieme, per cui non è necessaria interattività.

- gestione accessi, se la biblioteca dispone di un sistema di registrazione delle presenze;
- richiesta di servizi da parte dell'utente e modulo di erogazione servizi, che consente di registrare i diversi passi dell'iter del servizio;
- gestione del prestito locale e interbibliotecario di Polo;
- gestione del bilancio, per quanto concerne le acquisizioni bibliografiche;
- scarico di dati in formato UNIMARC.

Nonostante tutto, il progetto si scontrò nuovamente con una lentezza operativa, tanto che le prime fasi di test furono avviate solo nel corso del 2000.

Questo ennesimo ritardo portò, a partire dal 1999, ad un profondo ripensamento della struttura tecnologica di SBN, tanto che, grazie anche ad uno studio di fattibilità condotto fra il 1996 e il 1997 dalla società ETNOTEAM, nel 2002 venne avviato il progetto "Evoluzione dell'Indice SBN".

Gli obiettivi principali sono stati:

- il rinnovamento tecnologico dell'hardware e del software sia di base che applicativo con passaggio su piattaforma UNIX, utilizzo del protocollo di trasmissione TCP/IP e di middleware¹⁴⁷ standard di mercato, programmazione ad oggetti, utilizzo di XML, adozione dello standard UNICODE per la codifica dei caratteri;
- razionalizzazione, integrazione e ristrutturazione delle basi dati esistenti (libro moderno, antico e musica), separate e pertanto contenenti informazioni duplicate sui vari archivi; aggiunta di nuovi campi per rendere possibile la gestione di nuovi materiali (grafico, audiovisivo, dati elettronici); e offerta di nuovi servizi alle biblioteche integrando nell'unica base dati archivi di supporto quali authority files;
- apertura dell'Indice SBN a sistemi di gestione della biblioteca non SBN che utilizzino i più diffusi formati bibliografici (UNIMARC, MARC21) mediante

¹⁴⁷ Con "middleware" si intende quell'insieme di programmi che fungono da intermediari tra diverse applicazioni e componenti software. Esempi tipici di middleware sono il gestore del database (DBMS) o il web server.

la realizzazione di un'interfaccia standard di colloquio che consenta a sviluppatori di software di dotare altri applicativi del colloquio con l'Indice allargando così l'influenza e la distribuzione dei servizi offerti dal sistema centrale e la relativa utenza;

- gestione di livelli di cooperazione diversificati: ciascun Polo potrà scegliere il proprio livello di partecipazione al SBN (solo cattura dei dati, localizzazione del proprio posseduto, inserimento di nuove catalogazioni, correzione);
- sviluppo di nuove funzionalità, quali la catalogazione derivata (possibilità di importare dati bibliografici da altre basi dati) e la catalogazione a blocchi (riversamento in Indice di archivi bibliografici precedentemente costituiti, con confronto dei dati);
- sviluppo di funzioni di governo e monitoraggio sia delle prestazioni del sistema, sia dell'incremento delle basi dati, che saranno disponibili direttamente agli utenti anche periferici, ed elaborazione di strumenti di intercettazione dei duplicati¹⁴⁸.

Oltre al rinnovamento tecnologico, una svolta importante nel progetto è stata la decisione di aprire il dialogo con l'Indice anche a sistemi commerciali fin'ora estranei a SBN. Per farlo è stata scelta una completa aderenza agli standard nella definizione del nuovo protocollo di comunicazione SBN-MARC, completamente compatibile UNIMARC, scritto usando XML per la sintassi e codificato Unicode. Altro obiettivo importante è quello dell'uniformità catalografica, per cui viene reso obbligatorio l'allineamento dei record, ovvero l'aggiornamento all'ultima versione diffusa dall'Indice di un record corretto da un altro Polo/Biblioteca, prima di poter effettuare modifiche su quel record.

L'attenzione all'interoperabilità permette ora una maggiore autonomia, da parte delle biblioteche, nella scelta di adesione al Servizio, che offre ora una serie di funzioni strutturate su più livelli:

- ricerca: permette l'interrogazione della base dati e lo scarico dei record sulla

148 Cfr. ICCU[2006].

propria base dati locale;

- localizzazione: permette al Polo di comunicare all'Indice il possesso del documento;
- creazione: permette la catalogazione in Indice;
- modifica, cancellazione, fusione: permettono la manipolazione dei dati bibliografici, utilizzabile solo da chi possiede il record;
- richiesta allineamento: permette l'aggiornamento dei propri record sulla base delle modifiche effettuate anche dagli altri partner della rete. È possibile in due modalità: la prima prevede che l'Indice invii le modifiche di tutti i record localizzati presso il Polo/Biblioteca richiedente che siano stati corretti da parte di un altro Polo; la seconda permette di selezionare i dati da correggere sulla base di un arco temporale specifico ed eventualmente del tipo;
- comunicazione allineamento: permette al Polo/Biblioteca di comunicare all'Indice di aver ricevuto le modifiche quando il polo abbia scelto la prima modalità di allineamento¹⁴⁹.

Sulla base di queste funzioni sono stati individuati 4 livelli di conformità dei software che richiedono il dialogo con l'Indice:

1. ricerca;
2. ricerca e localizzazione per possesso;
3. ricerca e localizzazione per possesso, creazione e correzione dei record non condivisi;
4. ricerca, localizzazione per possesso e per gestione, catalogazione, correzione anche dei record condivisi e allineamento.

Ovviamente quasi tutti i software in circolazione sono conformi al primo livello, e permettono contestualmente anche lo scarico dei record sulla propria base dati locale.

¹⁴⁹ Cfr. Weston [2007].

Oltre al livello di dialogo con l'Indice, al momento della certificazione viene stabilito un diverso profilo sulla base della tipologia di materiale a cui si ha accesso. Il profilo minimo (obbligatorio) prevede l'accesso alle basi dati per Libro moderno, Libro antico e Authority Files degli Autori; in aggiunta sono presenti anche le basi dati di: Musica, Grafica e Cartografia per il tipo di materiale; titoli uniformi, marche, luoghi, soggetti, classi per gli Authority Files.

5.2. I sistemi integrati in Italia

La diffusione dei principali sistemi integrati utilizzati all'estero, non ha mai avuto un grande riscontro nel mercato italiano. Trattandosi spesso di sistemi adatti a realtà di medie-grandi dimensioni, e a fronte delle ridotte possibilità economiche di molte biblioteche, questi sono stati adottati prevalentemente da sistemi bibliotecari d'ateneo o grossi enti¹⁵⁰.

Inoltre, e forse è la causa principale, la presenza di SBN sul territorio nazionale¹⁵¹ e il ritardo nell'apertura al dialogo con applicativi non SBN, ha scoraggiato la scelta di applicativi non nativamente in grado di colloquiare con l'Indice¹⁵².

Anche se l'apertura ottenuta grazie al progetto "Evoluzione dell'Indice SBN" ha portato notevoli vantaggi sotto questo punto di vista, e anche in Italia si è diffusa la tendenza ad offrire i prodotti in modalità ASP¹⁵³, i vari produttori di sistemi integrati si sono visti maggiormente impegnati nella creazione di soluzioni per il dialogo con l'Indice, piuttosto che orientati allo sviluppo di nuove funzionalità come invece avvenuto all'estero¹⁵⁴.

In questo senso si nota un certo rallentamento a fornire risposte a quella tendenza alla dis-integrazione¹⁵⁵ che sta colpendo tutti i sistemi integrati diffusi negli anni '90.

150 Cfr. Bertini [2002]. Tra i prodotti diffusi nel nostro paese, soltanto Aleph 500 della Ex Libris ha una certa rilevanza in ambito internazionale, mentre quasi tutti gli altri prodotti sono stati sviluppati in Italia.

151 Le biblioteche aderenti a SBN rappresentato quasi un terzo del totale delle biblioteche in Italia.

152 Cfr. Bertini [2003].

153 Cfr. §2.3 Strumenti per l'integrazione

154 Cfr. Bertini [2009].

155 Cfr. §3.4. La dis-integrazione dei sistemi integrati.

Elenco dei principali sistemi integrati¹⁵⁶:

Aleph 500

<<http://www.exlibrisgroup.com/category/Aleph>>

Rivenditore: Ex Libris (Israele), distribuito in Italia da Atlantis.

Piattaforma: Unix/Linux, ASP.

Compatibilità con SBN: livello 4 e accesso alle basi Musica, Grafica, Cartografia, Titoli uniformi, Marche, Luoghi, Soggetti, Classi.

Oltre ai moduli di base, implementa quello per il prestito interbibliotecario. Implementa i principali standard (MARC, Z39.50, EDI, ISO-ILL, XML, OAI) e permette l'integrazione con i molti strumenti sviluppati da Ex Libris e acquistabili separatamente (p.e. Metalib, un metamotore di ricerche e la sua integrazione con SFX, un link resolver).

Amicus

<<http://www.primesource.it>>

Rivenditore: Prime Source (gruppo Visiant).

Piattaforma: Unix/Linux.

Compatibilità con SBN: livello 4 e accesso alle basi Musica, Grafica, Cartografia, Titoli uniformi, Marche, Luoghi, Soggetti, Classi.

Sono presenti i moduli di base per le funzioni di catalogazione, circolazione, acquisizione, OPAC e gestione periodici. Sono supportati i principali standard (MARC21, Z39.50, XML, EDI).

¹⁵⁶ La definizione di sistema integrato nel mercato italiano non è semplice, intanto per la difficoltà di reperire dati sulla diffusione dei prodotti, successivamente perché spesso il software implementa esclusivamente i moduli per catalogazione, circolazione e OPAC. Per questo si è fatto riferimento a Bertini [2009], AIB-SW [2008] e ai siti web dei rivenditori. Nei casi in cui non si siano riuscite a reperire informazioni sufficienti a poter definire il software "sistema integrato", questo è stato ommesso dall'elenco (p.e. @UOL.IT, Tecaweb), così come non vengono riportati dati su sistemi molto diffusi all'estero ma scarsamente utilizzati in Italia (p.e. Voyager, Millennium). Per quanto riguarda l'elenco di funzionalità o standard indicati come supportati nativamente, ci si è limitati a descrivere i principali e la mancanza in alcuni sistemi non significa necessariamente l'impossibilità di implementazione.

Bibliowin

<<http://www.bibliowin.net>>

Rivenditore: CG Soluzioni informatiche.

Piattaforma: Windows (versione 4.0), ASP (versione 5.0).

Compatibilità con SBN: livello 4 e accesso alle basi Titoli uniformi, Marche, Luoghi, Soggetti, Classi.

Sono state sviluppati i moduli base, compreso quello per il prestito interbibliotecario e la gestione del materiale elettronico, attivabili in modo opzionale a seconda della versione scelta. Supporta i principali standard (MARC, Z39.50, XML), RFID, ricerca federata e self-check.

EOS.Web

<<http://www.ifnet.it/eosweb>>

Rivenditore: EOS International (Stati Uniti), distribuito in Italia da Ifnet Srl.

Piattaforma: Windows, ASP.

Compatibilità con SBN: livello 3 e accesso alle basi Titoli uniformi, Marche, Soggetti, Classi.

Oltre ai moduli base e quello per il prestito interbibliotecario, permette l'integrazione con la gestione RFID, ricerca federata, self-check, gestione contenuti elettronici. Supporta i principali standard (MARC, Z39.50, XML).

Kentika

<<http://www.ifnet.it/alex/>>

Rivenditore: Kentika SAS (Francia), distribuito in Italia da Ifnet Srl.

Piattaforma: Windows, Macintosh, ASP.

Compatibilità con SBN: non prevista.

Sono implementati tutti i moduli base, ma quelli relativi a prestito, acquisizioni e periodici sono aggiuntivi. Supporta i principali standard (MARC, Z39.50, SRW, XML). Il precedente nome del prodotto era Alexandrie.

Libero

<<http://www.infologic.it/ita/libero.htm>>

Rivenditore: LIB-IT GmbH (Germania), distribuito in Italia da Infologic Srl

Piattaforma: Windows, Unix/Linux

Compatibilità con SBN: non prevista

Sono implementati tutti i moduli base e supporta i principali standard (MARC, Z39.50).

Sebina

<<http://www.sebina.it/index.html>>

Rivenditore: Data Management.

Piattaforma: Unix/Linux, Windows, ASP.

Compatibilità con SBN: livello 4 e accesso alle basi Musica, Grafica, Cartografia, Titoli uniformi, Marche, Luoghi, Soggetti, Classi.

Oltre ai moduli di base, fornisce anche quello per il prestito interbibliotecario e document delivery. Implementa i principali standard (MARC, Z39.50, EDI, ISO-ILL, XML, OAI, RFID, SIP2), inoltre offre la possibilità di arricchire l'OPAC e integrare diversi servizi di ricerca, tra cui un link resolver e un metamotore di ricerca.

SoSeBi TLM 4

<<http://www.rosebi.it>>

Rivenditore: SoSeBi.

Piattaforma: Windows.

Compatibilità con SBN: livello 3, accesso alla base dati minima (moderno, antico, authority autori).

Le funzionalità di base prevedono il modulo per la catalogazione, circolazione e gestione dei periodici. Supporta i principali standard (MARC, Z39.50, XML). È possibile l'integrazione con il modulo aggiuntivo per l'OPAC, fornito anche in modalità ASP.

Suite Nexus - Biblionauta

<http://www.nexusfi.it/BN_biblionauta.php>

Rivenditore: Nexus.

Piattaforma: Unix/Linux, ASP limitatamente a OPAC e catalogo.

Compatibilità con SBN: livello 4 e accesso alle basi Titoli uniformi, Marche, Luoghi, Soggetti, Classi.

Biblionauta è una suite di programmi separati integrabili tra loro, che comprendono: OPAC, catalogo, prestito, gestione acquisizione/periodici, modulo per il dialogo con Indice2.

Unibibliox

<<http://www.keyover.it/>>

Rivenditore: Keyover srl.

Piattaforma: Windows, Unix/Linux, ASP.

Compatibilità con SBN: livello 2, accesso alla base dati minima (moderno, antico, authority autori).

Sono presenti i moduli per le funzionalità base, inoltre è possibile l'integrazione con sistemi RFID.

Zetesis

Rivenditore: H&T.

Piattaforma: Windows.

Compatibilità con SBN: livello 3, accesso alla base dati minima (moderno, antico, authority autori).

Implementa i moduli per la circolazione, la catalogazione e l'OPAC. Inoltre permette la gestione delle postazioni ad accesso pubblico della biblioteca.

<<http://www.het.it/zetesis>>

5.3. Il software open source in Italia

La diffusione di software open source non è un fenomeno facile da monitorare. Quello che si può dire con certezza è che, come nel resto del mondo, anche in Italia il settore a maggiore diffusione è quello dell'IT. Questo perché le soluzioni tecnologiche maggiormente sviluppate, che hanno raggiunto, e alle volte superato, alternative commerciali, sono quelle relative alle infrastrutture informatiche come web server, database, sistemi operativi lato server, linguaggi di programmazione; sono ormai noti i casi di aziende, di diverse dimensioni, che investono nel software open source, sia in termini di sviluppo che in termini di utilizzo¹⁵⁷.

Per quanto riguarda la diffusione di applicativi lato client non si possono avere certezze, anche se sono parecchi i gruppi di interesse e le comunità di supporto nate intorno a progetti specifici¹⁵⁸.

Per quanto riguarda la Pubblica Amministrazione, una svolta importante è stata rappresentata dalla diffusione della Direttiva 19 Dicembre 2003¹⁵⁹, "Sviluppo ed utilizzazione dei programmi informatici da parte delle pubbliche amministrazioni", nota come "Direttiva Stanca", la quale, all'art. 4 (Criteri tecnici di comparazione), prevede che "Le pubbliche amministrazioni, nella predisposizione o nell'acquisizione dei programmi informatici, privilegiano le soluzioni che [...] basandosi su formati dei dati e interfacce aperte e standard, assicurino l'interoperabilità e la cooperazione applicativa tra i diversi sistemi informatici della pubblica amministrazione, salvo che ricorrano peculiari ed eccezionali esigenze di sicurezza e segreto; [...] che, in assenza di specifiche ragioni contrarie, rendano i sistemi informatici non dipendenti da un unico fornitore o da un'unica tecnologia proprietaria; la dipendenza è valutata tenendo conto dell'intera soluzione; [...] che esportino dati e documenti in più formati, di cui almeno uno di tipo aperto."

157 Cfr. Picerni – De Rossi [2009].

158 Ad esempio il gruppo PLUTO, nato a Padova nel 1992; oppure l'Italian Linux Society (ILS), nata nel 1994 per coordinare i vari Linux Users Group (LUG). Cfr. Picerni – De Rossi [2009].

159 Cfr. Frigimelica – Marchitelli [2009].

La stessa Direttiva, in linea con quanto già avvenuto nel resto dell'Unione Europea, ha previsto la costituzione dell'Osservatorio Open Source del Centro Nazionale per l'Informativa nella Pubblica Amministrazione (CNIPA), il quale funge da punto di riferimento a livello nazionale per quanto riguarda il tema dell'open source nelle P.A.

Per quanto riguarda gli applicativi specifici per le biblioteche, i dati e la letteratura di riferimento sono pochi¹⁶⁰ e si può dire che il fenomeno non abbia ancora visto un'ampia diffusione¹⁶¹.

Nel 1999 fu progettato lo sviluppo di un sistema integrato (DAFNE) per il sistema bibliotecario provinciale di Rovigo coordinato dall'Accademia dei Concordi, che prevedeva l'utilizzo di software open source (Linux, Apache, DBMS come MySQL o PostgreSQL, PHP), ma che purtroppo non fu mai portato a termine¹⁶².

Altri progetti per le biblioteche sviluppati in Italia sono: JOpac2, un OPAC sviluppato in Java e rilasciato con licenza GPL; PICO, un software per la costruzione di OPAC basato su WWWISIS (evoluzione di CDS/ISIS), sviluppato dal CILEA e rilasciato con licenza GPL.

Attualmente, un'apertura verso l'open source è rappresentata dalla licenza di Comperio, un'azienda che offre soluzioni per le biblioteche e rilascia la major release precedente del software distribuito alla biblioteca cliente che ne faccia richiesta, inoltre garantisce la diffusione del codice della release in uso in caso di scioglimento dell'azienda.

Inoltre, da alcuni mesi, il CILEA, che già partecipa intensamente alla comunità di Dspace¹⁶³, offre supporto commerciale per Koha e fornisce il suo contributo all'interno della comunità del software con la traduzione in italiano dell'OPAC e dell'interfaccia per il bibliotecario, e con il mantenimento della documentazione sul database MySQL¹⁶⁴.

160 Cfr. Sturman [2004], Baldi [2005], Di Giammarco [2005], Frigimelica - Marchitelli [2009].

161 Cfr. Frigimelica [2009].

162 Cfr. Sturman [2004].

163 Cfr. Bollini - Mornati [2009].

164 Cfr. Carassiti - Tajoli [2009].

A fronte di questi sforzi per la diffusione dell'open source, l'attenzione alle possibilità rappresentate soprattutto dai sistemi integrati di questo tipo è ancora molto bassa. Da un lato una certa diffidenza è rappresentata dalle problematiche relative alla connessione con SBN e dalla scarsità di case studies in merito, dall'altro è sicuramente dovuta ai timori circa il supporto commerciale per l'installazione e il mantenimento di tali sistemi, ancora carente in Italia.

Andrebbe comunque osservato che, come suggeriscono Frigimelica – Marchitelli [2009], la scelta open source non solo è una garanzia sul futuro, data dalla possibilità di accesso ai dati, ma rappresenta “una scelta di libertà, partecipazione, scambio, comunità, [...] autonomia: valori importanti, e senza vincoli.”.

6. Koha

6.1. Nascita ed evoluzione del software

Lo sviluppo di Koha è iniziato nel 1999, in Nuova Zelanda, da parte della Katipo Communications Ltd. per la Horowhenua Library Trust (HLT), la quale stava cercando un possibile sostituto al sistema in uso da 12 anni presso le biblioteche del distretto, che non fosse suscettibile ai possibili problemi relativi all'imminente cambio di data¹⁶⁵. Dopo un'attenta ricerca tra i prodotti in commercio, troppo costosi o non adeguati alle esigenze delle biblioteche, venne deciso di sviluppare in poco tempo un sistema integrato che non gravasse sull'infrastruttura a disposizione (quindi in grado di funzionare attraverso i modem dial up di una normale linea telefonica) e che potesse essere completamente a disposizione delle biblioteche, anche nel caso di fallimento della Katipo¹⁶⁶.

Il sistema, sviluppato secondo il paradigma LAMP¹⁶⁷ e rilasciato con licenza GPL, fu pronto poco dopo il passaggio al 2000 e ben presto fu riconosciuto come il primo sistema integrato open source in circolazione. Questo contribuì ad attirare l'interesse

¹⁶⁵ Il cosiddetto Millennium Bug (Y2K), per cui si temeva che i sistemi di datazione a due cifre dei calcolatori potessero subire problemi di ordinamento quando si fosse entrati nel 2000, non riconoscendo come valida la sequenza "...97, 98, 99, 00...". Anche se in realtà al cambio di data non avvennero problemi rilevanti, soprattutto grazie ad interventi di correzione preventivi, la questione sollevò diverse preoccupazioni.

¹⁶⁶ Cfr. Ransom [2009].

¹⁶⁷ Cfr. § 2.3. Strumenti per l'integrazione.

da parte di bibliotecari e sviluppatori, che andarono a formare una comunità internazionale impegnata fin da subito ad localizzare il prodotto. Una grossa spinta in questo senso arrivò nel 2002, quando la Nelsonville Public Library¹⁶⁸, decise di migrare a Koha. Per questo fu finanziato lo sviluppo del supporto per il formato MARC e del protocollo Z39.50, e nel 2003 migrò al sistema diventando la prima biblioteca americana ad utilizzare Koha.

Nel 2005, alcuni dipendenti della Nelsonville Public Library formarono LibLime, un'azienda impegnata nello sviluppo del software che iniziò a distribuire una versione commerciale di Koha dopo l'integrazione con l'applicativo per l'indicizzazione Zebra, della IndexData, il quale aumentò notevolmente le capacità del sistema in termini di scalabilità e performance¹⁶⁹.

La posizione di LibLime all'interno della comunità ha suscitato alcune controversie a partire da Settembre 2009, soprattutto a seguito di un cambio di atteggiamento nella modalità di rilascio del codice sviluppato dall'azienda, che ha deciso di dare priorità allo sviluppo della versione commerciale del prodotto, preferendo effettuare l'integrazione con la versione ufficiale solo alla fine del ciclo di produzione del software. Inoltre LibLime, che detiene i diritti sul marchio Koha negli Stati Uniti, sembra avere progressivamente ridotto la sua partecipazione alla comunità, tanto da creare notevoli difficoltà nell'aggiornamento del sito ufficiale che viene mantenuto sui loro server.

Per questo motivo la comunità, che attualmente è formata da diverse centinaia di biblioteche nel mondo e più di venti aziende coinvolte nello sviluppo e supporto del software, si sta muovendo verso la creazione di un'organizzazione in grado di garantire la continuità del progetto, tutelando il marchio Koha e fornendo ufficialmente gli strumenti per lo sviluppo. In questo senso ci si è espressi attraverso un sondaggio per rimettere la tutela del progetto nelle mani della Horowhenua Library Trust, attraverso la creazione di una commissione apposita¹⁷⁰. Nel frattempo

168 L'Athens County (Ohio), in cui è inserita la Nelsonville Public Library, serve un distretto di sette biblioteche, per un totale di 60.000 utenti e 250.000 items.

169 Cfr. Riewe [2008].

170 In questo senso, Biblibre ha passato a HLT le pratiche per la registrazione del marchio Koha in

è stato creato un apposito sito <<http://koha-community.org/>> che ospita le ultime release e gli aggiornamenti da parte della comunità.

6.2. Requisiti del sistema

Il sistema si basa su un'architettura di tipo LAMP. Ovvero su:

- una macchina con sistema operativo Linux;
- il web server Apache;
- un database MySQL¹⁷¹;
- il linguaggio di programmazione Perl.

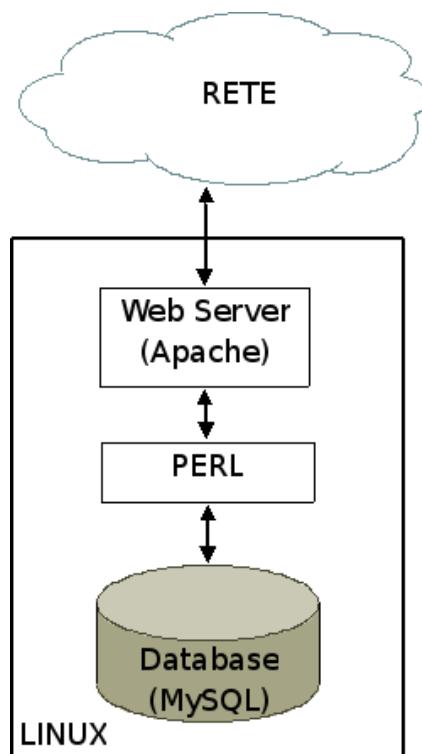


Figura 8: Sistema LAMP

Europa.

¹⁷¹ Il supporto per PostgreSQL è ancora in fase di sviluppo ed è previsto per la release 3.4.

La distribuzione di Linux consigliata è Debian, possibilmente una versione stabile (l'ultima release è la 5.0.4 Lenny). È comunque possibile utilizzare altre distribuzioni, ad esempio esistono installazioni su: Ubuntu, OpenSUSE, CentOS, Fedora, OpenBSD, Redhat, Slackware. Ma anche su Windows o MacOS X. Esistono inoltre pacchetti appositamente per virtual machines, chiamati Koha Virtual Appliances, utili soprattutto a scopo di testing.

Per ottenere Koha esistono principalmente due alternative: il download del pacchetto contenente l'ultima release del software dal sito ufficiale del progetto, oppure utilizzando il sistema Git per il controllo di versione¹⁷².

Una volta ottenuto il pacchetto applicativo è consigliabile preparare il sistema creando un utente apposito per l'amministrazione di Koha, a cui garantire i privilegi necessari a questo compito.

L'installazione di Perl (preferibilmente la versione 5.8) va integrata con alcuni moduli aggiuntivi presenti nel pacchetto di Koha, oppure, in alternativa, ottenuti e installati tramite CPAN.

Per quanto riguarda il database MySQL (preferibilmente la versione 5), questo va creato prima dell'installazione, associando all'utente amministratore di Koha i seguenti privilegi per la gestione: SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, ALTER, LOCK TABLES.

Una volta completata la preparazione del sistema operativo preferito, equipaggiato con Perl e MySQL, è opportuno installare anche i componenti della Index Data per Zebra, il motore dell'indicizzazione, e il web server Apache.

¹⁷² I sistemi per il controllo di versione, come Git, permettono la creazione di diversi repository pubblici online, sulla base delle diverse versioni, dal quale chiunque può attingere prelevando il codice sorgente. Una volta effettuate le modifiche in locale queste possono essere caricate sul repository e condivise con il resto degli sviluppatori.

6.3. L'installazione di Koha

L'installazione del pacchetto applicativo Koha comporta la compilazione del Makefile.PL, durante la quale verranno richieste varie informazioni, come: tipo di filesystem, informazioni per la connessione al database MySQL, informazioni per la connessione al database Zebra. Successivamente è possibile lanciare l'installazione.

Bisognerà poi configurare e avviare Apache e infine configurare e avviare i processi relativi a Zebra (si può decidere se utilizzare dei demoni o eseguirli quando necessario da riga di comando).

Finalmente è possibile collegarsi alla pagina web <http://<servername>:8080/> (dove servername è il nome del server su cui è installato Koha) e completare l'installazione tramite web, inserendo le preferenze di configurazione (che potranno essere cambiate successivamente dal menu di amministrazione) e dei dati d'esempio.

A questo punto, le uniche altre operazioni che sarà necessario eseguire da riga di comando sono legate a cronjobs, Zebra e altri script sviluppati come estensioni o tools a cui non corrisponde un template per l'interfaccia web, tra questi va segnalato lo script `bulkmarcimport.pl` per l'importazione batch di record MARC (più rapido rispetto alla versione via browser) e lo script `bulkauthimport.pl` per l'importazione batch di authority (al momento non presente via browser). Inoltre, se lo si desidera, sempre da riga di comando, è possibile installare diversi template sulla base della lingua scelta e di quelle che si intende rendere disponibili agli utenti.

Ovviamente anche le operazioni di aggiornamento o eventuale cancellazione del sistema comporteranno l'utilizzo di comandi da terminale, tuttavia il sistema è stato pensato per garantire un vasto numero di personalizzazioni direttamente dall'interfaccia web per lo staff.

6.4. *La struttura del software*

È possibile considerare Koha come logicamente strutturato su più livelli, a cui corrisponde una diversa collocazione dei file nel filesystem, sulla base dei compiti svolti e del livello di profondità rispetto alla percezione finale da parte dell'utilizzatore¹⁷³. I livelli individuabili sono quattro:

1. Il livello più basso e distante dall'utente è quello relativo alla memorizzazione dei dati ed è costituito dal database MySQL e dal database XML di Zebra. Il primo contiene i dati per la gestione del sistema, mentre il secondo si occupa di rappresentare le informazioni bibliografiche per indicizzare i record, sulla base del formato scelto in fase di installazione (MARC21 o UNIMARC).
2. Il secondo livello di astrazione è formato dai moduli perl (collocati nella cartella C4) per il colloquio con il database. Questi offrono una serie di funzionalità generiche richiamate di volta in volta dagli script di più alto livello.
3. Il terzo livello è composto dagli script perl che compiono la maggior parte delle operazioni richieste dal sistema. Sono di due tipi: da riga di comando Unix oppure script perl CGI (quindi accessibili via browser). A livello di filesystem sono collocati in cartelle dai nomi significativi, rispetto al modulo o alle funzionalità compiute: acqui, admin, authorities, basket, catalogue, cataloguing, circ, errors, installer, labels, misc, members, offline_circ, opac, reports, reserve, reviews, serials, sms, suggestion, tags, tools, virtualshelves.
4. L'ultimo livello è quello relativo agli script perl CGI e alle pagine HTML generate attraverso i templates. Queste utilizzano Javascript (basato sulle librerie YUI e jQuery), CSS e HTML. Sono proprio i templates (posizionati in koha-tmpl) che permettono la localizzazione del sistema, poiché si basano sulle diverse traduzioni dell'interfaccia disponibili (circa una cinquantina, a

173 Cfr. Carassiti – Tajoli [2009].

diversi livelli di completamento).

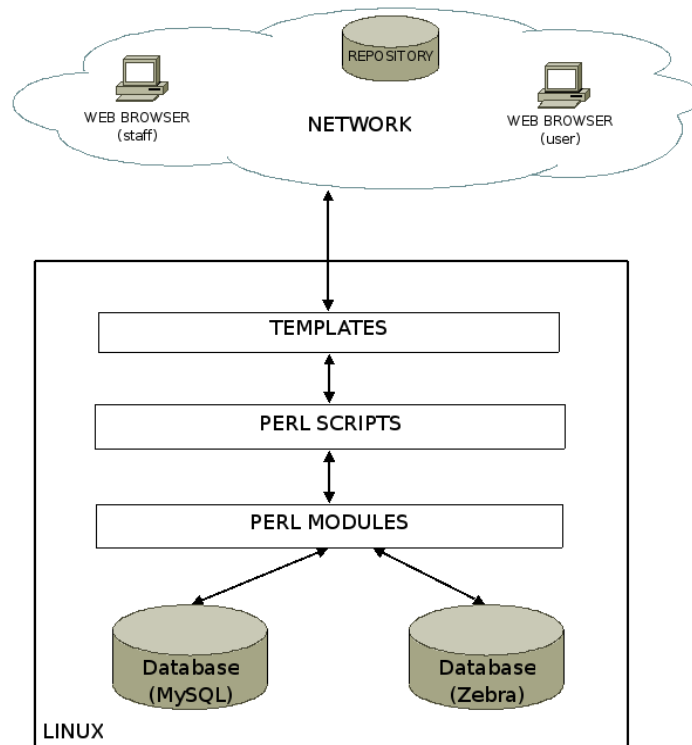


Figura 9: Struttura a livelli di Koha

L'interfaccia web del sistema è divisa in due ambienti: uno per gli utenti della biblioteca (OPAC) e uno per lo staff.

L'OPAC implementa le tipiche funzioni di un sistema integrato, come:

- ricerca per: titolo, autore, soggetto, serie, segnatura, ecc.;
- ricerca booleana;
- ricerca per liste;
- ricerca per tag definiti dagli utenti;
- possibilità di navigare tra i record;
- dettagli bibliografici;
- informazioni sulla disponibilità, con possibilità di rinnovo o prenotazione

del prestito;

- area personale dell'utente che comprende: visualizzazione dei dettagli utente, dei prestiti, delle multe, dei tag, definizione di liste e suggerimenti d'acquisto;
- contenuto accresciuto, ovvero la possibilità di arricchire i record con le informazioni provenienti per esempio da Amazon o Google Books;
- interfaccia multilingue;
- possibilità di personalizzazione di CSS, HTML e Javascript del sito web (p.e. aggiungendo messaggi da parte dello staff in diverse aree dello schermo).

L'interfaccia per lo staff, che può essere differenziata sulla base del profilo utente e dei permessi associati, è suddivisa in una serie di sezioni che raggruppano diverse funzionalità.

La sezione Amministrazione, tipicamente riservata al system librarian, comprende:

- Preferenze di Sistema Globali, ovvero le opzioni attivabili per l'amministrazione generale del sistema, le quali sono a loro volta divise nelle varie funzionalità offerte: Acquisizioni, Authority, Catalogazione, Circolazione, I18N/L10N (per la localizzazione), Log, OAI-PMH (sperimentale), OPAC, Utenti, Ricerca, Staff Client, Uso Locale.
- Parametri di base, permette di definire: le biblioteche che fanno parte del sistema, la gestione dei fondi e budgets, i tassi di cambio e i tipi di documenti (item) gestiti.
- Utenti e circolazione, permette di definire: categorie utente, tipologie di indirizzi, attributi degli utenti, regole di circolazione (p.e. ritardi, multe, limiti, avvisi).
- Catalogo, permette di definire: categorie e valori autorizzati, griglie di catalogazione, la mappatura tra il database di Koha e la struttura MARC, tipi

di authority, schemi di classificazione, regole di corrispondenza.

- Parametri aggiuntivi, permette di definire una lista di stopwords e l'elenco di server Z39.50 a cui è possibile connettersi.

La sezione Acquisizioni permette:

- la gestione dei suggerimenti d'acquisto;
- la definizione di budget e fondi (attraverso la sezione Amministrazione);
- la gestione dei fornitori;
- la gestione dell'ordine (creazione, modifica, ricezione, gestione dei ritardi).

La sezione Risorse in continuazione, per la gestione dei periodici, permette:

- creazione, gestione e rinnovo abbonamenti, con possibilità di avvisi automatici in caso di mancata ricezione;
- gestione solleciti;
- controllo abbonamenti in scadenza.

La sezione Catalogazione permette:

- importazione ed esportazione dati in formato MARC, con possibilità di mappare il record sulla base delle preferenze impostate;
- creazione di record bibliografici, anche derivati dalla connessione via Z39.50;
- creazione dei dati delle copie e gestione dei dettagli per l'oggetto;
- authority control.

La sezione Gestione utenti permette:

- la registrazione di utenti, con possibilità di gestirne diverse tipologie e lo storico;
- impostazione dei permessi per singolo utente;

La sezione Circolazione permette:

- la gestione della coda delle prenotazioni;
- il prestito, rinnovo, restituzione delle copie;
- gestione dei ritardi e delle multe;

La sezione Strumenti contiene diverse utility, come: importazione batch di record bibliografici, moderazione dei commenti utenti e dei tag, definizione dei messaggi, gestione delle news e del calendario, creazione di etichette.

La sezione Liste permette di definire diverse tipologie di liste: personali, pubbliche ma gestite dal creatore, libere ovvero gestibili da chiunque.

La sezione Reports fornisce la possibilità di creare, salvare ed esportare reports, sia guidati sulla base dei diversi moduli oppure a partire da query SQL.

6.5. Struttura del database

Le tabelle presenti nella versione 3.00.02 di Koha sono 116 ed è possibile suddividerle sulla base delle funzioni svolte.

Il database contiene sia dati gestionali per l'amministrazione del software che i record bibliografici e d'autorità.

Un diagramma Entità-Relazione di più alto livello, che tenga conto delle distinzioni per modulo, potrebbe essere rappresentato come in Figura 10.

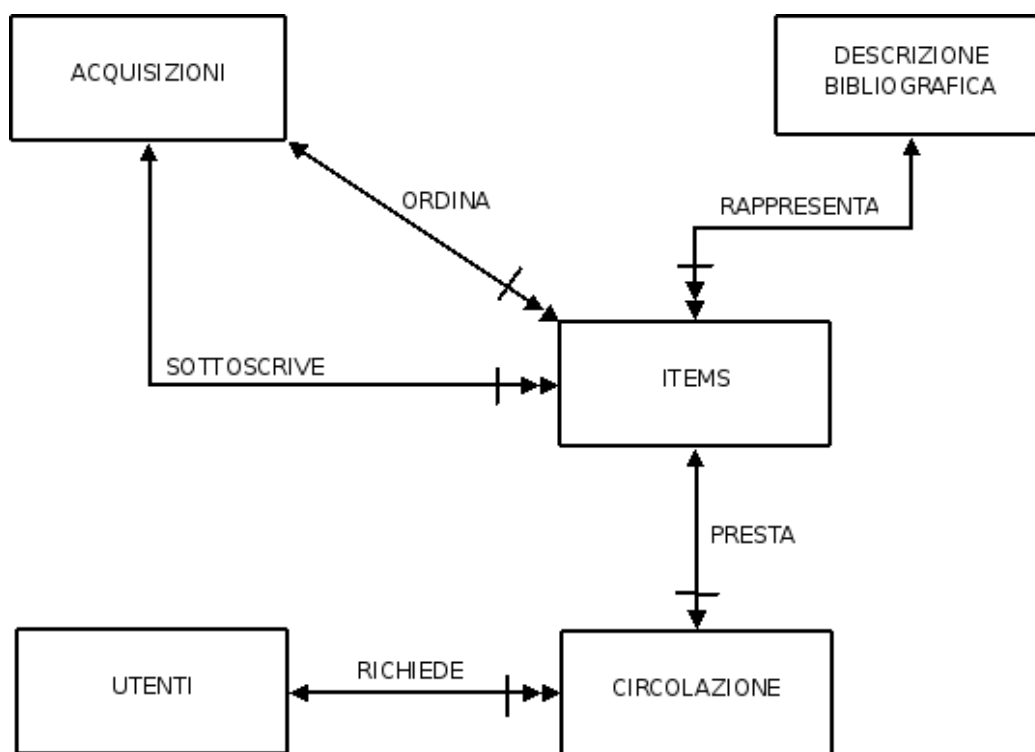


Figura 10: Modello E-R generalizzato per Koha 3.00.02

Il modulo acquisizioni può ordinare uno o più items, intesi come le copie fisiche degli oggetti posseduti dalla biblioteca. Inoltre, grazie al modulo periodici è possibile sottoscrivere uno o più abbonamenti, a cui corrisponderà la ricezione degli item relativi alla sottoscrizione. Un item può essere ordinato o sottoscritto attraverso una sola procedura di acquisizione o abbonamento.

Una descrizione bibliografica può rappresentare più items, mentre ad un item corrisponde una e una sola descrizione.

Un item può essere prestato attraverso il modulo per la circolazione e lo stesso modulo è utilizzato per il prestito di un item alla volta.

Un utente può effettuare una o più richieste di prestito, gestite dal modulo circolazione, mentre un certo prestito è richiesto da un solo utente.

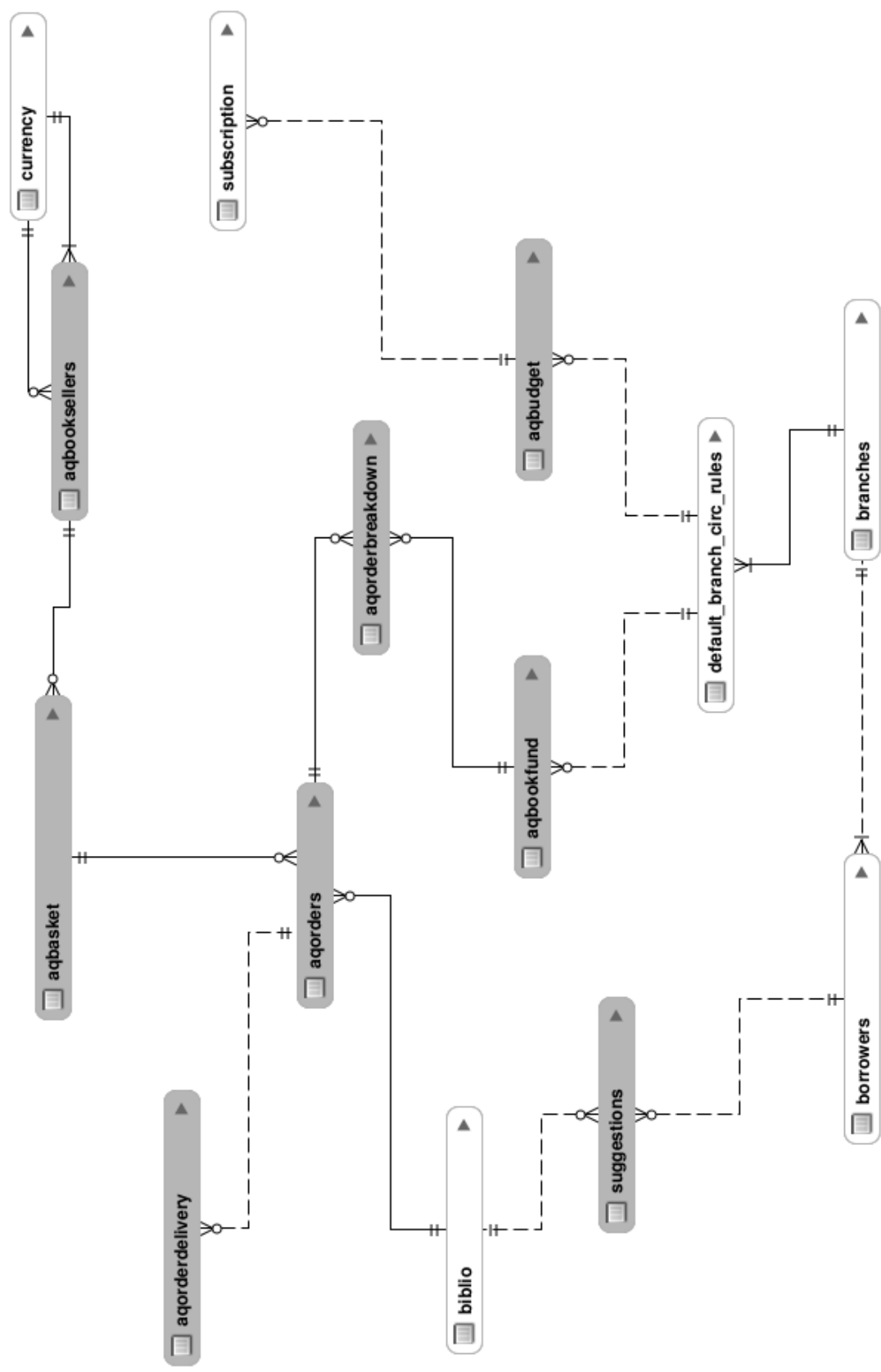
Mentre le entità Circolazione ed Utenti raggruppano effettivamente le tabelle relative, l'entità Acquisizioni comprende sia il modulo per le acquisizioni che il modulo per i periodici. Le entità Items e Descrizione Bibliografica raggruppano il

nutrito numero di tabelle relative alla catalogazione.

Restano escluse dalla generalizzazione le tabelle relative ai reports e quelle relative al funzionamento del sistema.

Di seguito viene riportato il diagramma E-R e la descrizione delle tabelle che compongono il database MySQL di Koha, suddivise in base ai principali moduli a cui fanno riferimento.

La descrizione è composta da: nome del campo, tipologia del campo, obbligatorietà. Gli indici sono indicati con una sottolineatura singola, le chiavi primarie con una sottolineatura doppia, le chiavi esterne con un asterisco.



6.5.1. Tabelle del modulo per le acquisizioni

aqbasket

(basketno INT(11) NOT NULL, creationdate DATE, closedate DATE, booksellerid* INT(11) NOT NULL, authorisedby VARCHAR(10), booksellerinvoicenumbers MEDIUMTEXT)

Descrizione: recipiente degli ordini di acquisto

aqbookfund

(bookfundid VARCHAR(10) NOT NULL, bookfundname MEDIUMTEXT, bookfundgroup VARCHAR(5), branchcode VARCHAR(10) NOT NULL)

Descrizione: fondi per le nuove acquisizioni.

aqbooksellers

(id INT(11) NOT NULL, name MEDIUMTEXT NOT NULL, address1 MEDIUMTEXT, address2 MEDIUMTEXT, address3 MEDIUMTEXT, address4 MEDIUMTEXT, phone VARCHAR(30), accountnumber MEDIUMTEXT, othersupplier MEDIUMTEXT, currency VARCHAR(3) NOT NULL, deliverydays SMALLINT(6), followupdays SMALLINT(6), followupcancel SMALLINT(6), specialty MEDIUMTEXT, booksellerfax MEDIUMTEXT, notes MEDIUMTEXT, bookselleremail MEDIUMTEXT, booksellerurl MEDIUMTEXT, contact VARCHAR(100), postal MEDIUMTEXT, url VARCHAR(255), contpos VARCHAR(100), contphone VARCHAR(100), confax VARCHAR(100), contaltphone VARCHAR(100), contemail VARCHAR(100), contnotes MEDIUMTEXT, active TINYINT(4), listprice* VARCHAR(10), invoiceprice* VARCHAR(10), gstreg TINYINT(4), listincgst TINYINT(4), invoiceincgst TINYINT(4), discount FLOAT, fax VARCHAR(50), nocalc INT(11), invoicedisc FLOAT)

Descrizione: informazioni sul rivenditore.

aqbudget

(bookfundid VARCHAR(10) NOT NULL, startdate DATE NOT NULL, enddate DATE, budgetamount DECIMAL(13,2), aqbudgetid INT(11) NOT NULL, branchcode VARCHAR(10))

Descrizione: informazioni sui budgets applicati ai fondi esistenti.

aqorderbreakdown

(ordernumber* INT(11), linenumber INT(11), branchcode VARCHAR(10), bookfundid* VARCHAR(10) NOT NULL, allocation SMALLINT(6))

Descrizione: dettagli sui fondi utilizzati per effettuare gli ordini.

aqorderdelivery

(ordernumber DATE, deliverynumber SMALLINT(6) NOT NULL, deliverydate VARCHAR(18), qtydelivered SMALLINT(6), deliverycomments MEDIUMTEXT NULL)

Descrizione: informazioni sulla consegna degli ordini.

aqorders

(ordernumber INT(11) NOT NULL, biblionumber* INT(11), title MEDIUMTEXT, entrydate DATE, quantity SMALLINT(6), currency VARCHAR(3), listprice DECIMAL(28,6), totalamount DECIMAL(28,6), datereceived DATE, booksellerinvoicenum MEDIUMTEXT, freight DECIMAL(28,6), unitprice DECIMAL(28,6), quantityreceived SMALLINT(6), cancelledby VARCHAR(10), datecancellationprinted DATE, notes MEDIUMTEXT, supplierreference MEDIUMTEXT, purchaseordernumber MEDIUMTEXT, subscription TINYINT(1), serialid VARCHAR(30), basketno* INT(11), biblioitemnumber INT(11), timestamp TIMESTAMP NOT NULL, rrp DECIMAL(13,2), ecost DECIMAL(13,2), gst DECIMAL(13,2), budgetdate DATE, sort1 VARCHAR(80), sort2 VARCHAR(80))

Descrizione: informazioni sugli ordini di acquisizione.

suggestions

(suggestionid INT(8) NOT NULL, suggestedby INT(11) NOT NULL, managedby INT(11), STATUS VARCHAR(10) NOT NULL, note MEDIUMTEXT, author VARCHAR(80), title VARCHAR(80), copyrightdate SMALLINT(6), publishercode VARCHAR(255), date TIMESTAMP NOT NULL, volumedesc VARCHAR(255), publicationyear SMALLINT(6), place VARCHAR(255), isbn VARCHAR(30), mailoverseeing SMALLINT(1), biblionumber INT(11), reason TEXT)

Descrizione: suggerimenti di acquisizione da parte degli utenti.

6.5.2. Tabelle del modulo per la catalogazione

auth_header

(authid BIGINT(20) UNSIGNED NOT NULL, authtypecode VARCHAR(10) NOT NULL, datecreated DATE, datemodified DATE, origincode VARCHAR(20), authtrees MEDIUMTEXT, marc BLOB, linkid BIGINT(20), marcxml LONGTEXT NOT NULL)

Descrizione: lista delle autorità in formato MARC

auth_subfield_structure

(authtypecode VARCHAR(10) NOT NULL, tagfield VARCHAR(3) NOT NULL, tagsubfield VARCHAR(1) NOT NULL, liblibrarian VARCHAR(255) NOT NULL, libopac VARCHAR(255) NOT NULL, repeatable TINYINT(4) NOT NULL, mandatory TINYINT(4) NOT NULL, tab TINYINT(1), authorised_value VARCHAR(10), value_builder VARCHAR(80), seealso VARCHAR(255), isurl TINYINT(1), hidden TINYINT(3) NOT NULL, linkid TINYINT(1) NOT NULL, kohafield VARCHAR(45), frameworkcode VARCHAR(8) NOT NULL)

Descrizione: sottocampi dei tag MARC usati per creare la creazione dell'authority list.

auth_tag_structure

(authtypecode* VARCHAR(10) NOT NULL, tagfield VARCHAR(3) NOT NULL, liblibrarian VARCHAR(255) NOT NULL, libopac VARCHAR(255) NOT NULL, repeatable TINYINT(4) NOT NULL, mandatory TINYINT(4) NOT NULL, authorised_value VARCHAR(10))

Descrizione: tag MARC usati per creare la creazione dell'authority list.

auth_types

(authtypecode* VARCHAR(10) NOT NULL, authtypetext VARCHAR(255) NOT NULL, auth_tag_to_report VARCHAR(3) NOT NULL, summary MEDIUMTEXT NOT NULL)

Descrizione: tipologie di authority MARC definibili.

authorised_values

(id INT(11) NOT NULL, category VARCHAR(10) NOT NULL, authorised_value VARCHAR(80) NOT NULL, lib VARCHAR(80), imageurl VARCHAR(200))

Descrizione: serie di valori autorizzati per la descrizione degli item.

biblio

(biblionumber INT(11) NOT NULL, frameworkcode VARCHAR(4) NOT NULL, author MEDIUMTEXT, title MEDIUMTEXT, unititle MEDIUMTEXT, notes MEDIUMTEXT, serial TINYINT(1), seriestitle MEDIUMTEXT, copyrightdate SMALLINT(6), timestamp TIMESTAMP NOT NULL, datecreated DATE NOT NULL, abstract MEDIUMTEXT)

Descrizione: descrizione bibliografica a livello di manifestazione.

biblio_framework

(frameworkcode VARCHAR(4) NOT NULL, frameworktext VARCHAR(255) NOT NULL)

Descrizione: framework bibliografico, utilizzato a scopi catalogafici.

biblioitems

(biblioitemnumber INT(11) NOT NULL, biblionumber* INT(11) NOT NULL, volume MEDIUMTEXT, number MEDIUMTEXT, itemtype VARCHAR(10), isbn VARCHAR(30), issn VARCHAR(9), publicationyear TEXT, publishercode VARCHAR(255), volumedate DATE, volumedesc TEXT, collectiontitle MEDIUMTEXT, collectionissn TEXT, collectionvolume MEDIUMTEXT, editionstatement TEXT, editionresponsibility TEXT, timestamp TIMESTAMP NOT NULL, illus VARCHAR(255), pages VARCHAR(255), notes MEDIUMTEXT, size VARCHAR(255), place VARCHAR(255), lcn VARCHAR(25), marc LONGBLOB, url VARCHAR(255), cn_source VARCHAR(10), cn_class VARCHAR(30), cn_item VARCHAR(10), cn_suffix VARCHAR(10), cn_sort VARCHAR(30), totalissues INT(10), marcxml LONGTEXT NOT NULL)

Descrizione: descrizione bibliografica associata agli item.

browser

(level INT(11) NOT NULL, classification VARCHAR(20) NOT NULL, description VARCHAR(255) NOT NULL, number BIGINT(20) NOT NULL, endnode TINYINT(4) NOT NULL)

Descrizione: contiene le definizioni delle categorie per la navigazione del catalogo.

class_sort_rules

(class_sort_rule VARCHAR(10) NOT NULL, description MEDIUMTEXT, sort_routine VARCHAR(30) NOT NULL)

Descrizione: regole di ordinamento per la classificazione.

class_sources

(cn_source VARCHAR(10) NOT NULL, description MEDIUMTEXT, used

TINYINT(4) NOT NULL, class_sort_rule* VARCHAR(10) NOT NULL)

Descrizione: fonti per la classificazione (p.e. DDC, UDC, LCC, ecc.) .

deletedbiblio

(biblionumber INT(11) NOT NULL, frameworkcode VARCHAR(4) NOT NULL, author MEDIUMTEXT, title MEDIUMTEXT, unititle MEDIUMTEXT, notes MEDIUMTEXT, serial TINYINT(1), seriestitle MEDIUMTEXT, copyrightdate SMALLINT(6), timestamp TIMESTAMP NOT NULL, datecreated DATE NOT NULL, abstract MEDIUMTEXT)

Descrizione: record cancellati delle informazioni bibliografiche. Tabella parallela a *biblio*.

deletedbiblioitems

(biblioitemnumber INT(11) NOT NULL, biblionumber INT(11) NOT NULL, volume MEDIUMTEXT, number MEDIUMTEXT, itemtype VARCHAR(10), isbn VARCHAR(30), issn VARCHAR(9), publicationyear TEXT, publishercode VARCHAR(255), volumedate DATE, volumedesc TEXT, collectiontitle MEDIUMTEXT, collectionissn TEXT, collectionvolume MEDIUMTEXT, editionstatement TEXT, editionresponsibility TEXT, timestamp TIMESTAMP NOT NULL, illus VARCHAR(255), pages VARCHAR(255), notes MEDIUMTEXT, size VARCHAR(255), place VARCHAR(255), lcn VARCHAR(25), marc LONGBLOB, url VARCHAR(255), cn_source VARCHAR(10), cn_class VARCHAR(30), cn_item VARCHAR(10), cn_suffix VARCHAR(10), cn_sort VARCHAR(30), totalissues INT(10), marcxml LONGTEXT NOT NULL)

Descrizione: records cancellati delle informazioni bibliografiche sulle copie (p.e. raccolte, collezioni ecc.). Tabella parallela a *biblioitems*.

deleteditems

(itemnumber INT(11) NOT NULL, biblionumber INT(11) NOT NULL, biblioitemnumber INT(11) NOT NULL, barcode VARCHAR(20), dateaccessioned DATE, booksellerid MEDIUMTEXT, homebranch VARCHAR(10), price DECIMAL(8,2), replacementprice DECIMAL(8,2), replacementpricedate DATE, datelastborrowed DATE, datelastseen DATE, stack TINYINT(1), notforloan TINYINT(1) NOT NULL, damaged TINYINT(1) NOT NULL, itemlost TINYINT(1) NOT NULL, withdrawn TINYINT(1) NOT NULL, itemcallnumber VARCHAR(30), issues SMALLINT(6), renewals SMALLINT(6), reserves SMALLINT(6), restricted TINYINT(1), itemnotes MEDIUMTEXT, holdingbranch VARCHAR(10), paidfor MEDIUMTEXT, timestamp TIMESTAMP NOT NULL, location VARCHAR(80), onloan DATE, cn_source VARCHAR(10), cn_sort VARCHAR(30), ccode VARCHAR(10), materials VARCHAR(10), uri VARCHAR(255), itype VARCHAR(10), more_subfields_xml LONGTEXT, enumchron VARCHAR(80), copynumber VARCHAR(32), marc LONGBLOB)

Descrizione: records degli item cancellati. Tabella parallela a *items*.

items

(itemnumber INT(11) NOT NULL, biblionumber INT(11) NOT NULL, biblioitemnumber* INT(11) NOT NULL, barcode VARCHAR(20), dateaccessioned DATE, booksellerid MEDIUMTEXT, homebranch* VARCHAR(10), price DECIMAL(8,2), replacementprice DECIMAL(8,2), replacementpricedate DATE, datelastborrowed DATE, datelastseen DATE, stack TINYINT(1), notforloan TINYINT(1) NOT NULL, damaged TINYINT(1) NOT NULL, itemlost TINYINT(1) NOT NULL, withdrawn TINYINT(1) NOT NULL, itemcallnumber VARCHAR(30), issues SMALLINT(6), renewals SMALLINT(6), reserves SMALLINT(6), restricted TINYINT(1), itemnotes MEDIUMTEXT, holdingbranch* VARCHAR(10), paidfor MEDIUMTEXT, timestamp TIMESTAMP NOT NULL, location VARCHAR(80), onloan DATE, cn_source VARCHAR(10), cn_sort VARCHAR(30), ccode VARCHAR(10), materials VARCHAR(10), uri VARCHAR(255), itype VARCHAR(10), more_subfields_xml LONGTEXT, enumchron VARCHAR(80), copynumber VARCHAR(32))

Descrizione: items posseduti dalla biblioteca.

itemtypes

(itemtype VARCHAR(10) NOT NULL, description MEDIUMTEXT, renewalsallowed SMALLINT(6), rentalcharge DOUBLE, notforloan SMALLINT(6), imageurl VARCHAR(200), summary TEXT)

Descrizione: tipologie di item e informazioni relative.

labels

(labelid INT(11) NOT NULL, batch_id VARCHAR(10) NOT NULL, itemnumber VARCHAR(100) NOT NULL, timestamp TIMESTAMP NOT NULL)

Descrizione: etichette per la stampa (per utenti, libri).

labels_conf

(id INT(4) NOT NULL, barcodetype CHAR(100), title INT(1), subtitle INT(1), itemtype INT(1), barcode INT(1), dewey INT(1), classification INT(1), subclass INT(1), itemcallnumber INT(1), author INT(1), issn INT(1), isbn INT(1), startlabel INT(2) NOT NULL, printingtype CHAR(32), formatstring VARCHAR(64), layoutname CHAR(20) NOT NULL, guidebox INT(1), active TINYINT(1), fonttype CHAR(10) CHARACTER SET utf8 COLLATE utf8_unicode_ci, ccode CHAR(4) CHARACTER SET utf8 COLLATE utf8_unicode_ci, callnum_split INT(1), text_justify CHAR(1) CHARACTER SET utf8 COLLATE utf8_unicode_ci)

Descrizione: set di configurazioni per le etichette.

labels_profile

(tmpl_id INT(4) NOT NULL, prof_id INT(4) NOT NULL)

Descrizione: associazione tra i template delle etichette e i profili di stampa.

labels_templates

(tmpl_id INT(4) NOT NULL, tmpl_code CHAR(100), tmpl_desc CHAR(100), page_width FLOAT, page_height FLOAT, label_width FLOAT, label_height FLOAT, topmargin FLOAT, leftmargin FLOAT, cols INT(2), rows INT(2), colgap FLOAT, rowgap FLOAT, active INT(1), units CHAR(20), fontsize INT(4) NOT NULL, font CHAR(10) NOT NULL)

Descrizione: informazioni sulle etichette stampabili per gli items catalogati.

marc_matchers

(matcher_id INT(11) NOT NULL, code VARCHAR(10) NOT NULL, description VARCHAR(255) NOT NULL, record_type VARCHAR(10) NOT NULL, threshold INT(11) NOT NULL)

Descrizione: matchers MARC per l'identificazione dei duplicati.

marc_subfield_structure

(tagfield VARCHAR(3) NOT NULL, tagsubfield VARCHAR(1) CHARACTER SET utf8 COLLATE utf8_bin NOT NULL, liblibrarian VARCHAR(255) NOT NULL, libopac VARCHAR(255) NOT NULL, repeatable TINYINT(4) NOT NULL, mandatory TINYINT(4) NOT NULL, kohafield VARCHAR(40), tab TINYINT(1), authorised_value VARCHAR(20), authtypecode VARCHAR(20), value_builder VARCHAR(80), isurl TINYINT(1), hidden TINYINT(1), frameworkcode VARCHAR(4) NOT NULL, seealso VARCHAR(1100), link VARCHAR(80), defaultvalue TEXT)

Descrizione: struttura dei sottocapi MARC utilizzata per la catalogazione.

marc_tag_structure

(tagfield VARCHAR(3) NOT NULL, liblibrarian VARCHAR(255) NOT NULL, libopac VARCHAR(255) NOT NULL, repeatable TINYINT(4) NOT NULL, mandatory TINYINT(4) NOT NULL, authorised_value VARCHAR(10), frameworkcode VARCHAR(4) NOT NULL)

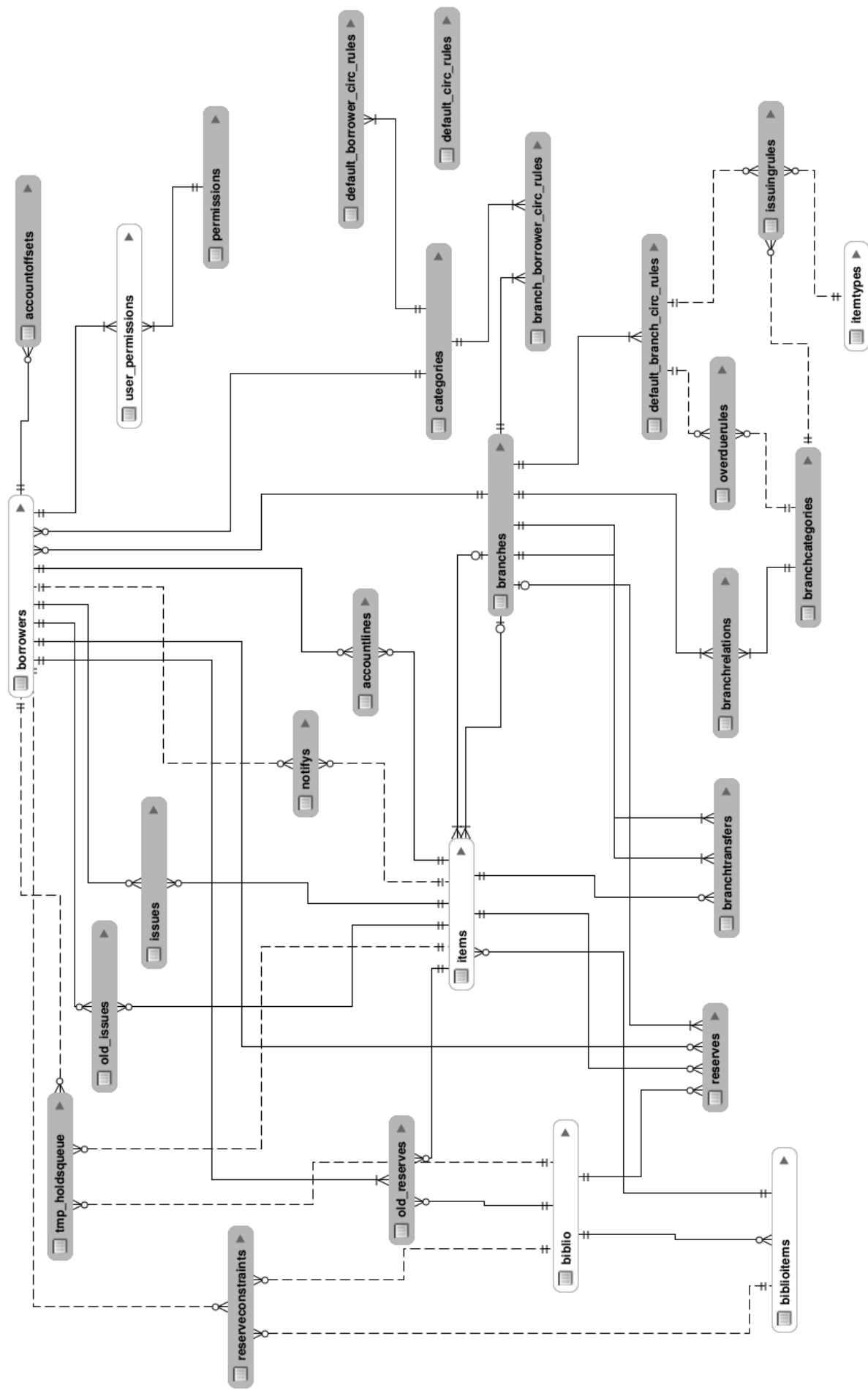
Descrizione: struttura dei campi MARC utilizzata per la catalogazione.

printers_profile

(prof_id INT(4) NOT NULL, printername VARCHAR(40) NOT NULL, tmpl_id* INT(4) NOT NULL, paper_bin VARCHAR(20) NOT NULL, offset_horz FLOAT,

offset_vert FLOAT, creep_horz FLOAT, creep_vert FLOAT, unit CHAR(20) NOT NULL)

Descrizione: associa le stampanti ai profili di stampa.



6.5.3. Tabelle del modulo per la circolazione

accountlines

(borrowernumber* INT(11) NOT NULL, accountno SMALLINT(6) NOT NULL, itemnumber* INT(11), date DATE, amount DECIMAL(28,6), description MEDIUMTEXT, dispute MEDIUMTEXT, accounttype VARCHAR(5), amountoutstanding DECIMAL(28,6), lastincrement DECIMAL(28,6), timestamp TIMESTAMP NOT NULL, notify_id INT(11) NOT NULL, notify_level INT(2) NOT NULL)

Descrizione: mette in relazione *borrowers* e *items*, tenendo conto delle operazioni relative al prestito.

accountoffsets

(borrowernumber* INT(11) NOT NULL, accountno SMALLINT(6) NOT NULL, offsetaccount SMALLINT(6) NOT NULL, offsetamount DECIMAL(28,6), timestamp TIMESTAMP NOT NULL)

Descrizione: fornisce informazioni sui pagamenti e sull'aumento delle multe a carico dell'utente.

branch_borrower_circ_rules

(branchcode* VARCHAR(10) NOT NULL, categorycode* VARCHAR(10) NOT NULL, maxissueqty INT(4))

Descrizione: regola di circolazione sulla base della sottobiblioteca e categoria utente.

branchcategories

(categorycode VARCHAR(10) NOT NULL, categoryname VARCHAR(32), codedescription MEDIUMTEXT, categorytype VARCHAR(16))

Descrizione: categorie a cui può appartenere una sottobiblioteca.

branches

(branchcode VARCHAR(10) NOT NULL, branchname MEDIUMTEXT NOT NULL, branchaddress1 MEDIUMTEXT, branchaddress2 MEDIUMTEXT, branchaddress3 MEDIUMTEXT, branchphone MEDIUMTEXT, branchfax MEDIUMTEXT, branchemail MEDIUMTEXT, issuing TINYINT(4), branchip VARCHAR(15), branchprinter VARCHAR(100))

Descrizione: sottobiblioteche di una biblioteca.

branchrelations

(branchcode* VARCHAR(10) NOT NULL, categorycode* VARCHAR(10) NOT NULL)

Descrizione: relazione molti a molti tra sottobiblioteche e categorie.

branchtransfers

(itemnumber* INT(11) NOT NULL, datesent DATETIME, frombranch* VARCHAR(10) NOT NULL, datearrived DATETIME, tobranch* VARCHAR(10) NOT NULL, comments MEDIUMTEXT)

Descrizione: items trasferiti da una sottobiblioteca a un'altra.

default_borrower_circ_rules

(categorycode* VARCHAR(10) NOT NULL, maxissueqty INT(4))

Descrizione: regole di circolazione sulla base della categoria utente.

default_branch_circ_rules

(branchcode* VARCHAR(10) NOT NULL, maxissueqty INT(4))

Descrizione: regole di circolazione sulla base della sottobiblioteca.

default_circ_rules

(singleton ENUM(singleton) NOT NULL, maxissueqty INT(4))

Descrizione: regole di circolazione predefinite.

issues

(borrowernumber* INT(11), itemnumber* INT(11), date_due DATE, branchcode VARCHAR(10), issuingbranch VARCHAR(18), returndate DATE, lastreneweddate DATE, return VARCHAR(4), renewals TINYINT(4), timestamp TIMESTAMP NOT NULL, issuedate DATE))

Descrizione: prestiti correnti a carico dell'utente.

issuingrules

(categorycode VARCHAR(10) NOT NULL, itemtype VARCHAR(10) NOT NULL, restrictedtype TINYINT(1), rentaldiscount DECIMAL(28,6), reservecharge DECIMAL(28,6), fine DECIMAL(28,6), finedays INT(11), firstremind INT(11), chargeperiod INT(11), accountsent INT(11), chargename VARCHAR(100), maxissueqty INT(4), issuelength INT(4), branchcode VARCHAR(10) NOT NULL)

Descrizione: regole del prestito (per il calcolo automatico delle multe, ecc.).

notifys

(notify_id INT(11) NOT NULL, borrowernumber INT(11) NOT NULL, itemnumber INT(11) NOT NULL, notify_date DATE, notify_send_date DATE, notify_level INT(1) NOT NULL, method VARCHAR(20) NOT NULL)

Descrizione: notificazioni mandate all'utente.

old_issues

(borrowernumber* INT(11), itemnumber* INT(11), date_due DATE, branchcode VARCHAR(10), issuingbranch VARCHAR(18), returndate DATE, lastreneweddate DATE, return VARCHAR(4), renewals TINYINT(4), timestamp TIMESTAMP NOT NULL, issuedate DATE)

Descrizione: archivio dei prestiti. Tabella parallela a *issues*.

old_reserves

(borrowernumber* INT(11), reservedate DATE, biblionumber* INT(11), constrainttype VARCHAR(1), branchcode VARCHAR(10), notificationdate DATE, reminderdate DATE, cancellationdate DATE, reservenotes MEDIUMTEXT, priority SMALLINT(6), found VARCHAR(1), timestamp TIMESTAMP NOT NULL, itemnumber* INT(11), waitingdate DATE)

Descrizione: archivio delle prenotazioni. Tabella parallela a *reserves*.

overduerules

(branchcode VARCHAR(10) NOT NULL, categorycode VARCHAR(10) NOT NULL, delay1 INT(4), letter1 VARCHAR(20), debarred1 VARCHAR(1), delay2 INT(4), debarred2 VARCHAR(1), letter2 VARCHAR(20), delay3 INT(4), letter3 VARCHAR(20), debarred3 INT(1))

Descrizione: regole per i ritardi.

permissions

(module_bit* INT(11) NOT NULL, code VARCHAR(30) NOT NULL, description VARCHAR(255))

Descrizione: permessi utenti raggruppati per *userflags*.

reserveconstraints

(borrowernumber INT(11) NOT NULL, reservedate DATE, biblionumber INT(11) NOT NULL, biblioitemnumber INT(11), timestamp TIMESTAMP NOT NULL)

Descrizione: per gestire prenotazioni con particolari vincoli.

reserves

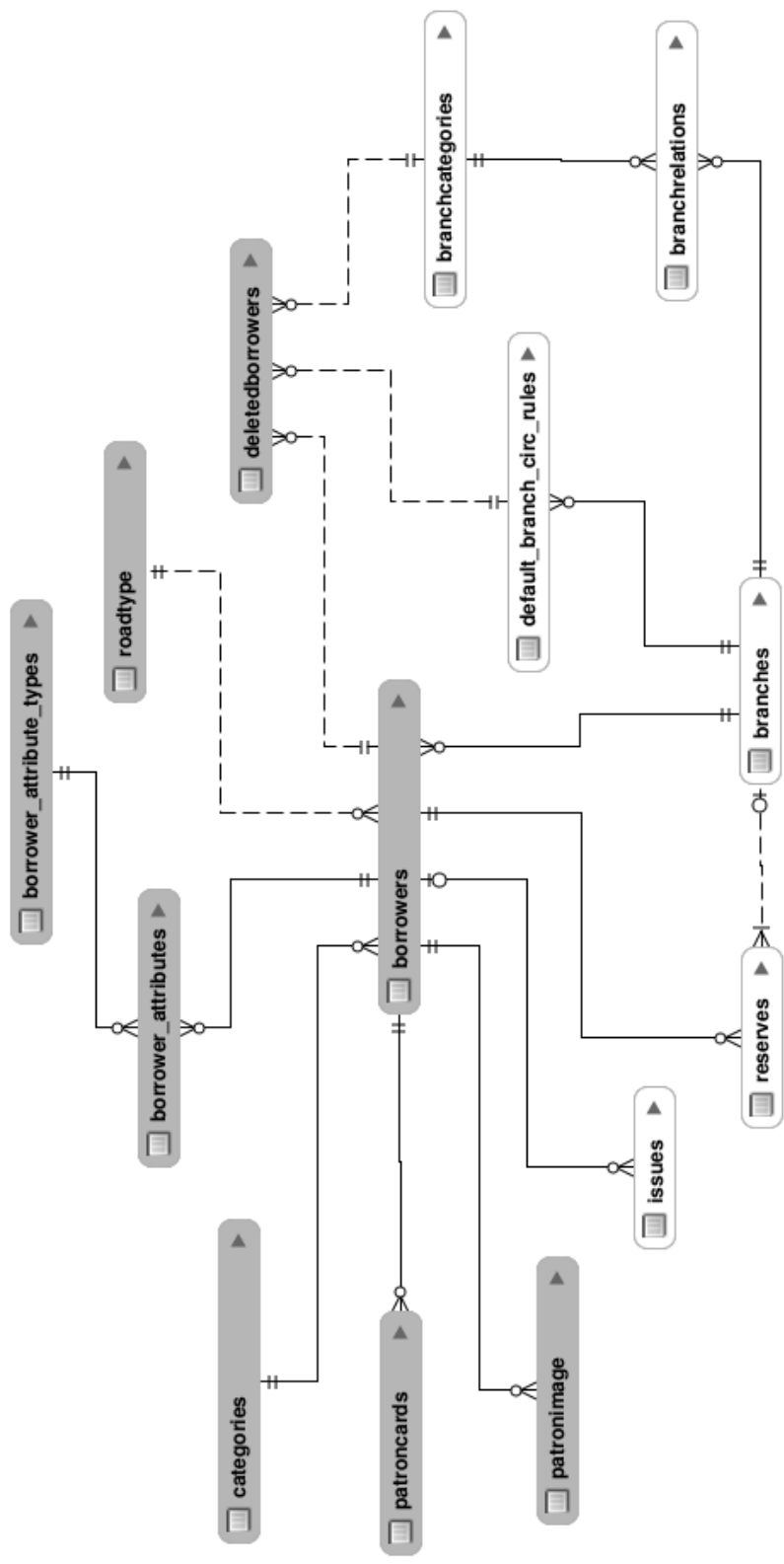
(borrowernumber* INT(11) NOT NULL, reservedate DATE, biblionumber* INT(11) NOT NULL, constrainttype VARCHAR(1), branchcode* VARCHAR(10), notificationdate DATE, reminderdate DATE, cancellationdate DATE, reservenotes MEDIUMTEXT, priority SMALLINT(6), found VARCHAR(1), timestamp TIMESTAMP NOT NULL, itemnumber* INT(11), waitingdate DATE)

Descrizione: gestisce le prenotazioni sugli items.

tmp_holdsqueue

(biblionumber INT(11), itemnumber INT(11), barcode VARCHAR(20), surname MEDIUMTEXT NOT NULL, firstname TEXT, phone TEXT, borrowernumber INT(11) NOT NULL, cardnumber VARCHAR(16), reservedate DATE, title MEDIUMTEXT, itemcallnumber VARCHAR(30), holdingbranch VARCHAR(10), pickbranch VARCHAR(10), notes TEXT NULL)

Descrizione: coda per gestire le prenotazioni.



6.5.4. Tabelle del modulo per la gestione utenti

borrower_attribute_types

(code VARCHAR(10) NOT NULL, description VARCHAR(255) NOT NULL, repeatable TINYINT(1) NOT NULL, unique_id TINYINT(1) NOT NULL, opac_display TINYINT(1) NOT NULL, password_allowed TINYINT(1) NOT NULL, staff_searchable TINYINT(1) NOT NULL, authorised_value_category VARCHAR(10))

Descrizione: lista di tipi di attributi estesi applicabile all'utente.

borrower_attributes

(borrowernumber* INT(11) NOT NULL, code* VARCHAR(10) NOT NULL, attribute VARCHAR(64), password VARCHAR(64))

Descrizione: attributi estesi associati all'utente.

borrowers

(borrowernumber INT(11) NOT NULL, cardnumber VARCHAR(16), surname MEDIUMTEXT NOT NULL, firstname TEXT, title MEDIUMTEXT, othernames MEDIUMTEXT, initials TEXT, streetnumber VARCHAR(10), streettype VARCHAR(50), address MEDIUMTEXT NOT NULL, address2 TEXT, city MEDIUMTEXT NOT NULL, zipcode VARCHAR(25), email MEDIUMTEXT, phone TEXT, mobile VARCHAR(50), fax MEDIUMTEXT, emailpro TEXT, phonepro TEXT, B_streetnumber VARCHAR(10), B_streettype VARCHAR(50), B_address VARCHAR(100), B_city MEDIUMTEXT, B_zipcode VARCHAR(25), B_email TEXT, B_phone MEDIUMTEXT, dateofbirth DATE, branchcode* VARCHAR(10) NOT NULL, categorycode* VARCHAR(10) NOT NULL, dateenrolled DATE, dateexpiry DATE, gonenoaddress TINYINT(1), lost TINYINT(1), debarred TINYINT(1), contactname MEDIUMTEXT, contactfirstname TEXT, contacttitle TEXT, guarantorid INT(11), borrowernotes MEDIUMTEXT, relationship VARCHAR(100), ethnicity VARCHAR(50), ethnotes VARCHAR(255), sex VARCHAR(1), password VARCHAR(30), flags INT(11), userid VARCHAR(30), opacnote MEDIUMTEXT, contactnote VARCHAR(255), sort1 VARCHAR(80), sort2 VARCHAR(80), altcontactfirstname VARCHAR(255), altcontactsurname VARCHAR(255), altcontactaddress1 VARCHAR(255), altcontactaddress2 VARCHAR(255), altcontactaddress3 VARCHAR(255), altcontactzipcode VARCHAR(50), altcontactphone VARCHAR(50), smsalertnumber VARCHAR(50)) Tabelle relative al modulo per la circolazione:

Descrizione: informazioni sull'utente, sia personali che per la biblioteca.

categories

(categorycode VARCHAR(10) NOT NULL, description MEDIUMTEXT, enrolmentperiod SMALLINT(6), upperagelimit SMALLINT(6), dateofbirthrequired TINYINT(1), finetype VARCHAR(30), bulk TINYINT(1), enrolmentfee DECIMAL(28,6), overduenoticerrequired TINYINT(1), issuelimit SMALLINT(6), reservefee DECIMAL(28,6), category_type VARCHAR(1) NOT NULL)

Descrizione: categorie di utenti (p.e. studente, docente, staff, ecc.).

deletedborrowers

(borrowernumber INT(11) NOT NULL, cardnumber VARCHAR(9) NOT NULL, surname MEDIUMTEXT NOT NULL, firstname TEXT, title MEDIUMTEXT, othernames MEDIUMTEXT, initials TEXT, streetnumber VARCHAR(10), streettype VARCHAR(50), address MEDIUMTEXT NOT NULL, address2 TEXT, city MEDIUMTEXT NOT NULL, zipcode VARCHAR(25), email MEDIUMTEXT, phone TEXT, mobile VARCHAR(50), fax MEDIUMTEXT, emailpro TEXT, phonepro TEXT, B_streetnumber VARCHAR(10), B_streettype VARCHAR(50), smsalertnumber VARCHAR(50), B_address VARCHAR(100), B_city MEDIUMTEXT, B_zipcode VARCHAR(25), B_email TEXT, B_phone MEDIUMTEXT, dateofbirth DATE, branchcode VARCHAR(10) NOT NULL, categorycode VARCHAR(10), dateenrolled DATE, dateexpiry DATE, gonenoaddress TINYINT(1), lost TINYINT(1), debarred TINYINT(1), contactname MEDIUMTEXT, contactfirstname TEXT, contacttitle TEXT, guarantorid INT(11), borrowernotes MEDIUMTEXT, relationship VARCHAR(100), ethnicity VARCHAR(50), ethnotes VARCHAR(255), sex VARCHAR(1), password VARCHAR(30), flags INT(11), userid VARCHAR(30), opacnote MEDIUMTEXT, contactnote VARCHAR(255), sort1 VARCHAR(80), sort2 VARCHAR(80), altcontactfirstname VARCHAR(255), altcontactsurname VARCHAR(255), altcontactaddress1 VARCHAR(255), altcontactaddress2 VARCHAR(255), altcontactaddress3 VARCHAR(255), altcontactzipcode VARCHAR(50), altcontactphone VARCHAR(50))

Descrizione: records degli utenti cancellati. Tabella parallela a *borrowers*.

patroncards

(cardid INT(11) NOT NULL, batch_id VARCHAR(10) NOT NULL, borrowernumber* INT(11) NOT NULL, timestamp TIMESTAMP NOT NULL)

Descrizione: tessere utenti.

patronimage

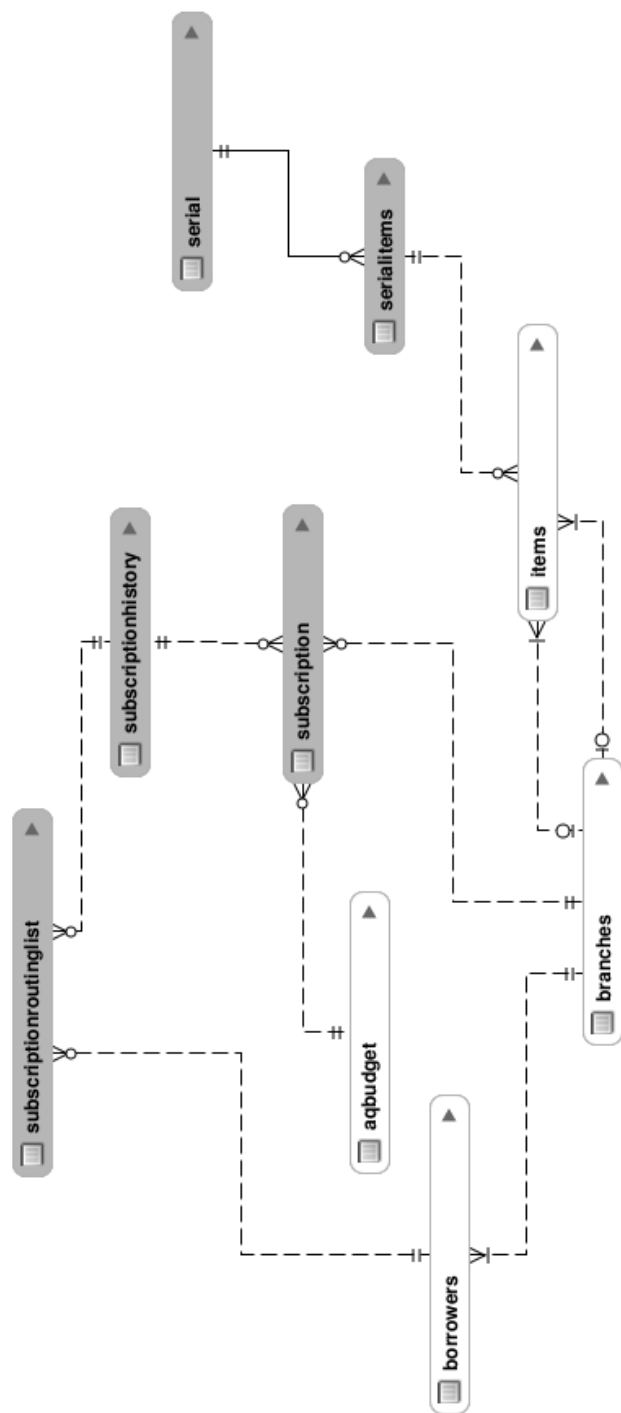
(cardnumber* VARCHAR(16) NOT NULL, mimetype VARCHAR(15) NOT NULL, imagefile MEDIUMBLOB NOT NULL)

Descrizione: immagini associate ai tessere utenti.

roadtype

(roadtypeid INT(11) NOT NULL, road_type VARCHAR(100) NOT NULL)

Descrizione: lista dei tipi di indirizzi accettati (p.e. via, piazza, ecc.).



6.5.5. Tabelle del modulo per la gestione dei periodici

serial

(serialid INT(11) NOT NULL, biblionumber VARCHAR(100) NOT NULL, subscriptionid VARCHAR(100) NOT NULL, serialseq VARCHAR(100) NOT NULL, status TINYINT(4) NOT NULL, planneddate DATE, notes TEXT, publisheddate DATE, itemnumber TEXT, claimdate DATE, routingnotes TEXT)

Descrizione: informazioni sui periodici.

serialitems

(itemnumber INT(11) NOT NULL, serialid* INT(11) NOT NULL)

Descrizione: items di un periodico.

subscription

(biblionumber INT(11) NOT NULL, subscriptionid INT(11) NOT NULL, librarian VARCHAR(100), startdate DATE, aqbooksellerid INT(11), cost INT(11), aqbudgetid INT(11), weeklength INT(11), monthlength INT(11), numberlength INT(11), periodicity TINYINT(4), dow VARCHAR(100), numberingmethod VARCHAR(100), notes MEDIUMTEXT, status VARCHAR(100) NOT NULL, add1 INT(11), every1 INT(11), whenmorethan1 INT(11), setto1 INT(11), lastvalue1 INT(11), add2 INT(11), every2 INT(11), whenmorethan2 INT(11), setto2 INT(11), lastvalue2 INT(11), add3 INT(11), every3 INT(11), innerloop1 INT(11), innerloop2 INT(11), innerloop3 INT(11), whenmorethan3 INT(11), setto3 INT(11), lastvalue3 INT(11), issuesatonce TINYINT(3) NOT NULL, firstacquidate DATE, manualhistory TINYINT(1) NOT NULL, irregularity TEXT, letter VARCHAR(20), numberpattern TINYINT(3), distributedto TEXT, internalnotes LONGTEXT, callnumber TEXT, branchcode VARCHAR(10) NOT NULL, hemisphere TINYINT(3), lastbranch VARCHAR(10), serialsadditems TINYINT(1) NOT NULL)

Descrizione: informazioni sugli abbonamenti ai periodici.

subscriptionhistory

(biblionumber INT(11) NOT NULL, subscriptionid INT(11) NOT NULL, histstartdate DATE, enddate DATE, missinglist LONGTEXT NOT NULL, recievedlist LONGTEXT NOT NULL, opacnote VARCHAR(150) NOT NULL, librariannote VARCHAR(150) NOT NULL)

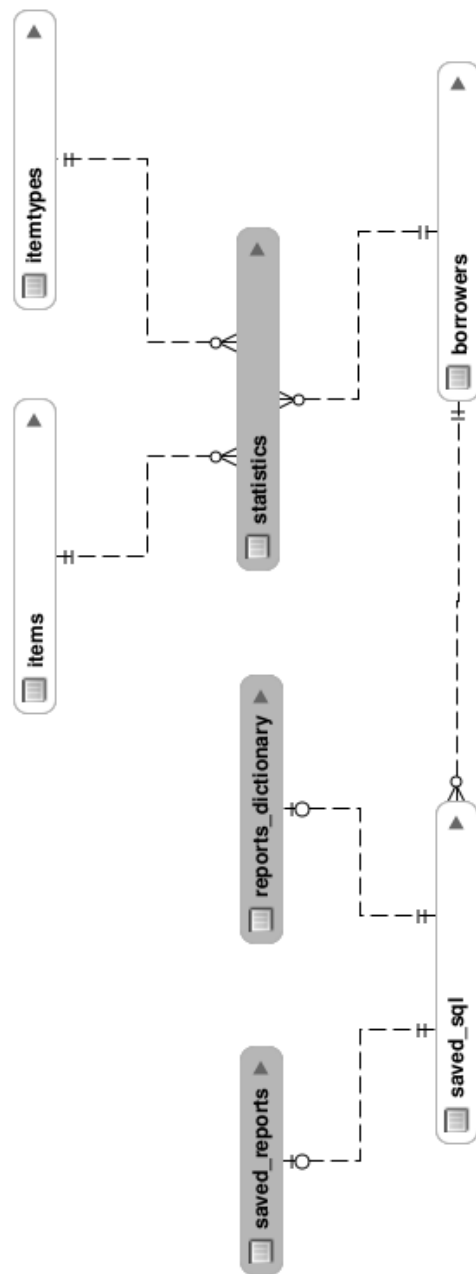
Descrizione: storico degli abbonamenti.

subscriptionroutinglist

(routingid INT(11) NOT NULL, borrowernumber INT(11), ranking INT(11),

subscriptionid INT(11))

Descrizione: routing list degli abbonamenti ai periodici degli utenti.



6.5.6. Tabelle relative ai reports

reports_dictionary

(id INT(11) NOT NULL, name VARCHAR(255), description TEXT, date_created DATETIME, date_modified DATETIME, saved_sql TEXT, area INT(11))

Descrizione: dizionario di criteri personali per i reports.

saved_reports

(id INT(11) NOT NULL, report_id INT(11), report LONGTEXT, date_run DATETIME)

Descrizione: archivio dei reports salvati.

statistics

(datetime DATETIME, branch VARCHAR(10), proccode VARCHAR(4), value DOUBLE, type VARCHAR(16), other MEDIUMTEXT, usercode VARCHAR(10), itemnumber INT(11), itemtype VARCHAR(10), borrowernumber INT(11), associatedborrower INT(11))

Descrizione: statistiche.

6.5.7. Tabelle di sistema

action_logs

(action_id INT(11) NOT NULL, timestamp TIMESTAMP NOT NULL, user INT(11) NOT NULL, module TEXT, action TEXT, object INT(11), info TEXT)

Descrizione: log delle azioni compiute sul sistema.

alert

(alertid INT(11) NOT NULL, borrowernumber* INT(11) NOT NULL, type VARCHAR(10) NOT NULL, externalid VARCHAR(20) NOT NULL)

Descrizione: solleciti legati al beneficiario del prestito

borrower_message_preferences

(borrower_message_preference_id INT(11) NOT NULL, borrowernumber* INT(11) NOT NULL, message attribute id* INT(11), days_in_advance INT(11), wants_digest TINYINT(1) NOT NULL)

Descrizione: preferenze dell'utente sulla ricezione dei messaggi.

borrower_message_transport_preferences

(borrower_message_preference_id* INT(11) NOT NULL, message_transport_type* VARCHAR(20) NOT NULL)

Descrizione: preferenze dell'utente sulla comunicazione dei messaggi.

cities

(cityid INT(11) NOT NULL, city_name VARCHAR(100) NOT NULL, city_zipcode VARCHAR(20))

Descrizione: informazioni sulle città.

currency

(currency VARCHAR(10) NOT NULL, symbol VARCHAR(5), timestamp TIMESTAMP NOT NULL, rate FLOAT)

Descrizione: informazioni sulle valute utilizzate.

ethnicity

(code* VARCHAR(10) NOT NULL, name VARCHAR(255))

Descrizione: nomi etnici.

import_batches

(import_batch_id INT(11) NOT NULL, matcher_id INT(11), template_id INT(11),
branchcode VARCHAR(10), num_biblios INT(11) NOT NULL, num_items INT(11)
NOT NULL, upload_timestamp TIMESTAMP NOT NULL, overlay_action
ENUM(replace,create_new,use_template,ignore) NOT NULL, nomatch_action
ENUM(create_new,ignore) NOT NULL, item_action
ENUM(always_add,add_only_for_matches,add_only_for_new,ignore) NOT NULL,
import_status
ENUM(staging,staged,importing,imported,reverting,reverted,cleaned) NOT NULL,
batch_type ENUM(batch,z3950) NOT NULL, file_name VARCHAR(100), comments
MEDIUMTEXT)

Descrizione: informazioni sulle operazioni per l'importazione batch di records.

import_biblios

(import_record_id* INT(11) NOT NULL, matched_biblionumber INT(11),
control_number VARCHAR(25), original_source VARCHAR(25), title
VARCHAR(128), author VARCHAR(80), isbn VARCHAR(30), issn VARCHAR(9),
has_items TINYINT(1) NOT NULL)

Descrizione: record bibliografici associati a un'importazione di gruppo (batch).

import_items

(import_items_id INT(11) NOT NULL, import_record_id* INT(11) NOT NULL,
itemnumber INT(11), branchcode VARCHAR(10), status
ENUM(error,staged,imported,reverted,ignored) NOT NULL, marcxml LONGTEXT
NOT NULL, import_error MEDIUMTEXT)

Descrizione: items associati a un'importazione batch.

import_record_matches

(import_record_id* INT(11) NOT NULL, candidate_match_id INT(11) NOT NULL,
score INT(11) NOT NULL)

Descrizione: controlla la corrispondenza di records per eventuali duplicazioni.

import_records

(import_record_id INT(11) NOT NULL, import_batch_id* INT(11) NOT NULL,
branchcode VARCHAR(10), record_sequence INT(11) NOT NULL,
upload_timestamp TIMESTAMP NOT NULL, import_date DATE, marc
LONGBLOB NOT NULL, marcxml LONGTEXT NOT NULL, marcxml_old
LONGTEXT NOT NULL, record_type ENUM(biblio,auth,holdings) NOT NULL,
overlay_status ENUM(no_match,auto_match>manual_match,match_applied) NOT
NULL, status ENUM(error,staged,imported,reverted,items_reverted,ignored) NOT

NULL, import_error MEDIUMTEXT, encoding VARCHAR(40) NOT NULL, z3950random VARCHAR(40))

Descrizione: importazione batch di records.

language_descriptions

(subtag VARCHAR(25), type VARCHAR(25), lang VARCHAR(25), description VARCHAR(255))

Descrizione: descrizione delle lingue.

language_rfc4646_to_iso639

(rfc4646_subtag VARCHAR(25), iso639_2_code VARCHAR(25))

Descrizione: mappatura degli identificatori delle lingue RFC4646 sui codici ISO639.

language_script_bidi

(rfc4646_subtag VARCHAR(25), bidi VARCHAR(3))

Descrizione: script di identificatori RFC4646 bidirezionali associati alle lingue.

language_script_mapping

(language_subtag VARCHAR(25), script_subtag VARCHAR(25))

Descrizione: mappatura degli identificatori e script dei tag RFC4646.

language_subtag_registry

(subtag VARCHAR(25), type VARCHAR(25), description VARCHAR(25), added DATE)

Descrizione: registro dei tag delle lingue RFC 4646.

letter

(module VARCHAR(20) NOT NULL, code VARCHAR(20) NOT NULL, name VARCHAR(100) NOT NULL, title VARCHAR(200) NOT NULL, content TEXT)

Descrizione: template per le lettere utilizzate nei messaggi agli utenti.

matchchecks

(matcher_id* INT(11) NOT NULL, matchcheck_id INT(11) NOT NULL, source_matchpoint_id* INT(11) NOT NULL, target_matchpoint_id* INT(11) NOT NULL)

Descrizione: regola di matching (per l'importazione batch).

matcher_matchpoints

(matcher_id* INT(11) NOT NULL, matchpoint_id* INT(11) NOT NULL)

Descrizione: matchers MARC associati ai matchpoints.

matchpoint_component_norms

(matchpoint_component_id INT(11) NOT NULL, sequence INT(11) NOT NULL, norm_routine VARCHAR(50) NOT NULL)

Descrizione: componenti della normalizzazione dei matchpoints.

matchpoint_components

(matchpoint_id* INT(11) NOT NULL, matchpoint_component_id INT(11) NOT NULL, sequence INT(11) NOT NULL, tag VARCHAR(3) NOT NULL, subfields VARCHAR(40) NOT NULL, offset INT(4) NOT NULL, length INT(4) NOT NULL)

Descrizione: componenti dei matchpoints.

matchpoints

(matcher_id* INT(11) NOT NULL, matchpoint_id INT(11) NOT NULL, search_index VARCHAR(30) NOT NULL, score INT(11) NOT NULL)

Descrizione: regole di matching (per l'importazione batch).

message_attributes

(message_attribute_id INT(11) NOT NULL, message_name VARCHAR(20) NOT NULL, takes_days TINYINT(1) NOT NULL)

Descrizione: attributi dei messaggi per gli utenti.

message_queue

(message_id INT(11) NOT NULL, borrowernumber* INT(11), subject TEXT, content TEXT, message_transport_type* VARCHAR(20) NOT NULL, status ENUM(sent,pending,failed,deleted) NOT NULL, time_queued TIMESTAMP NOT NULL, to_address MEDIUMTEXT, from_address MEDIUMTEXT, content_type TEXT)

Descrizione: coda per la trasmissione dei messaggi all'utente.

message_transport_types

(message_transport_type VARCHAR(20) NOT NULL)

Descrizione: tipologia di comunicazione del messaggio (p.e. mail, sms, ecc.)

message_transports

(message_attribute_id* INT(11) NOT NULL, message_transport_type* VARCHAR(20) NOT NULL, is_digest TINYINT(1) NOT NULL, letter_module* VARCHAR(20) NOT NULL, letter_code VARCHAR(20) NOT NULL)

Descrizione: associa le informazioni sulla consegna dei messaggi alle lettere.

nozebra

(server VARCHAR(20) NOT NULL, indexname VARCHAR(40) NOT NULL, value VARCHAR(250) NOT NULL, biblionumbers LONGTEXT NOT NULL)

Descrizione: definizioni di indicizzazione usate da Koha quando Zebra non è attivato.

opac_news

(idnew INT(10) UNSIGNED NOT NULL, title VARCHAR(250) NOT NULL, new TEXT NOT NULL, lang VARCHAR(25) NOT NULL, timestamp TIMESTAMP NOT NULL, expirationdate DATE, number INT(11))

Descrizione: archivio delle news dell'OPAC.

printers

(printername VARCHAR(40) NOT NULL, printqueue VARCHAR(20), printtype VARCHAR(20))

Descrizione: informazioni sulle stampanti.

repeatable_holidays

(id INT(11) NOT NULL, branchcode VARCHAR(10) NOT NULL, weekday SMALLINT(6), day SMALLINT(6), month SMALLINT(6), title VARCHAR(50) NOT NULL, description TEXT NOT NULL)

Descrizione: giorni di chiusura ricorrenti (non speciali).

reviews

(reviewid INT(11) NOT NULL, borrowernumber INT(11), biblionumber INT(11), review TEXT, approved TINYINT(4), datereviewed DATETIME)

Descrizione: recensioni scritte dagli utenti sugli items.

saved_sql

(id INT(11) NOT NULL, borrowernumber INT(11), date_created DATETIME, last_modified DATETIME, savedsql TEXT, last_run DATETIME, report_name VARCHAR(255), type VARCHAR(255), notes TEXT)

Descrizione: archivia valori di queries SQL.

services_throttle

(service_type VARCHAR(10) NOT NULL, service_count VARCHAR(45))

Descrizione: contatore di accessi ai servizi (reset giornaliero).

sessions

(id VARCHAR(32) NOT NULL, a_session TEXT NOT NULL)

Descrizione: archivio di session ID.

special_holidays

(id INT(11) NOT NULL, branchcode VARCHAR(10) NOT NULL, day SMALLINT(6) NOT NULL, month SMALLINT(6) NOT NULL, year SMALLINT(6) NOT NULL, isexception SMALLINT(1) NOT NULL, title VARCHAR(50) NOT NULL, description TEXT NOT NULL)

Descrizione: giorni di chiusura speciali (non ripetibili)

stopwords

(word VARCHAR(255) NULL)

Descrizione: elenco delle stopwords (usato per migliorare la ricerca).

systempreferences

(variable VARCHAR(50) NOT NULL, value TEXT, options MEDIUMTEXT, explanation TEXT, type VARCHAR(20))

Descrizione: preferenze di sistema.

tags

(entry VARCHAR(255) NOT NULL, weight BIGINT(20) NOT NULL)

Descrizione: tag definiti dall'utente.

tags_all

(tag_id INT(11) NOT NULL, borrowernumber* INT(11) NOT NULL, biblionumber* INT(11) NOT NULL, term VARCHAR(255) NOT NULL, language INT(4), date_created DATETIME NOT NULL)

Descrizione: associazione tra tag e item che tiene traccia del creatore.

tags_approval

(term VARCHAR(255) NOT NULL, approved INT(1) NOT NULL, date_approved DATETIME, approved_by* INT(11), weight_total INT(9) NOT NULL)

Descrizione: informazioni sull'approvazione del tag.

tags_index

(term* VARCHAR(255) NOT NULL, biblionumber* INT(11) NOT NULL, weight INT(9) NOT NULL)

Descrizione: tags indicizzati sulla base della descrizione bibliografica a cui fanno riferimento.

user_permissions

(borrowernumber* INT(11) NOT NULL, module_bit* INT(11) NOT NULL, code* VARCHAR(30))

Descrizione: permessi utente.

userflags

(bit INT(11) NOT NULL, flag VARCHAR(30), flagdesc VARCHAR(255), defaulton INT(11))

Descrizione: flags per i permessi utente.

virtualshelfcontents

(shelfnumber* INT(11) NOT NULL, biblionumber* INT(11) NOT NULL, flags INT(11), dateadded TIMESTAMP NOT NULL)

Descrizione: associa le descrizioni bibliografiche agli scaffali virtuali.

virtualshelves

(shelfnumber INT(11) NOT NULL, shelfname VARCHAR(255), owner VARCHAR(80), category VARCHAR(1), sortfield VARCHAR(16), lastmodified TIMESTAMP NOT NULL)

Descrizione: scaffali virtuali per organizzare i records bibliografici.

z3950servers

(host VARCHAR(255), port INT(11), db VARCHAR(255), userid VARCHAR(255), password VARCHAR(255), name MEDIUMTEXT, id INT(11) NOT NULL, checked SMALLINT(6), rank INT(11), syntax VARCHAR(80), icon TEXT, position ENUM(primary,secondary) NOT NULL, type ENUM(zed,opensearch) NOT NULL, encoding TEXT, description TEXT NOT NULL)

Descrizione: elenco di servers Z39.50 per l'information retrieval.

zebraqueue

(id INT(11) NOT NULL, biblio_auth_number BIGINT(20) UNSIGNED NOT NULL, operation CHAR(20) NOT NULL, server CHAR(20) NOT NULL, done INT(11) NOT NULL, time TIMESTAMP NOT NULL)

Descrizione: coda utilizzata dal sistema di indicizzazione Zebra.

6.6. La comunità e gli strumenti di sviluppo

La longevità del sistema è sicuramente dovuta anche alla presenza di una forte comunità, distribuita a livello internazionale, che comprende più di cento sviluppatori, oltre alle migliaia di utilizzatori del sistema (staff e utenti).

Gli strumenti utilizzati comprendono:

- Il sito ufficiale del progetto <<http://koha-community.org/>> che raccoglie tutte le informazioni relative alla diffusione, alle news dei rivenditori, alla documentazione, alle versioni demo del sistema.
- Alcuni siti nazionali e di gruppi di interesse (p.e. <<http://www.koha-fr.org/>> per la comunità francofona, <<http://kudos.koha.org/>> per la comunità di utenti e sviluppatori dagli Stati Uniti, <<http://www.koha.it/>> per la comunità italiana).
- Diverse mailing lists: Koha-announce, per gli annunci di nuove release; Koha-br, per la comunità brasiliana; Koha-bugs, per l'attività relativa al database dei bugs; koha-commits, per i commits a Koha; Koha-devel, per gli sviluppatori; Koha-i18n-german, per gli utenti tedeschi; Koha-patches, per le patches; Koha-pt, per la comunità portoghese; Koha-translate, per le comunicazioni relative alle traduzioni; Koha-win32, per il port di Koha su Windows; Koha-zebra, per l'integrazione tra Koha e Zebra.
- Un wiki per la condivisione e creazione di vario materiale e documentazione utile (case studies, guide, report di meeting, pagine per gruppi locali, ecc.).
- La piattaforma <<http://translate.koha.org/>> basata sul software Pootle, per gestire e condividere le varie traduzioni.
- La piattaforma <<http://bugs.koha.org/>> basata su Bugzilla per la segnalazione e gestione dei bugs.
- Il sistema per il controllo di versione Git, accessibile dalla pagina <<http://git.koha.org/>>.
- Diversi meeting tenuti nel canale IRC ufficiale.

Le decisioni sullo sviluppo del progetto sono di norma prese durante incontri ufficiali tenuti via IRC a cui può partecipare chiunque, inoltre sono precedute e fatte seguire da comunicazioni via mailing lists per permettere un'approfondita discussione per ogni tema di rilievo ed ottenere il più ampio consenso tra la comunità.

Alla luce degli ultimi avvenimenti circa il sito ufficiale del progetto, non più aggiornato da LibLime, sono stati presi dei provvedimenti per il ritorno al patrocinio da parte dell'Horowhenua Library Trust, nell'attesa che un'apposita Commissione decida il futuro assetto organizzativo da dare al progetto¹⁷⁴.

Per quanto riguarda l'Italia, è stato predisposto il sito <<http://www.koha.it/>> su DokuWiki per l'aggiornamento sugli sviluppi della situazione italiana, tutt'ora mantenuto prevalentemente dagli sviluppatori presso il CILEA.

¹⁷⁴ Al momento le scelte predominanti sono tra la creazione di un'apposita fondazione (in modo simile a quanto successo per il progetto Apache), o l'adesione ad un'organizzazione più ampia.

7. L'ipotesi Koha a Ca' Foscari

7.1. Il contesto di riferimento

Il Sistema delle Biblioteche dell'Ateneo fornisce a tutto l'Ateneo i servizi per accedere alle informazioni e alle fonti documentarie necessarie per lo studio e la ricerca; garantisce la conservazione, sviluppa l'acquisizione e gestisce la fruizione del patrimonio bibliotecario e documentale dell'Ateneo¹⁷⁵.

Il Sistema è formato dalle seguenti biblioteche:

- Biblioteca di Area Economica (BEC), che comprende la Biblioteca "G. Luzzatto" a S. Giobbe e le biblioteche afferenti del Dipartimento di Scienze Giuridiche a Ca' Bottacin, del Dipartimento di Matematica Applicata a Ca' Dolfin e del Corso di laurea in Economia e Gestione dei Servizi Turistici-EGeST a Oriago.
- Biblioteca di Area Linguistica (BALI), che comprende le cinque biblioteche di Dipartimento: Americanistica, Iberistica e Slavistica, Scienze del Linguaggio, Studi Euroasiatici, Studi europei e post-coloniali, Studi sull'Asia orientale.
- Biblioteca di Area Scientifica (BAS).
- Biblioteca di Area Umanistica (BAUM).
- Biblioteca di Servizio Didattico (BSD).

¹⁷⁵ Cfr. Unive-SBA [2009a]

- Altre biblioteche e collezioni speciali:
 - Fondo Storico di Ateneo e Archivio Tesi
 - la biblioteca del Centro di Documentazione Europea-CDE
 - la biblioteca del Centro Linguistico Interfacoltà-CLI
 - la biblioteca del Centro interuniversitario di studi veneti-CISVE

Inoltre fanno parte del Sistema il Servizio Amministrativo del Sistema (SASBA), che si occupa degli aspetti amministrativo-contabili e i Servizi Generali Bibliotecari (SeGeBi) che curano la gestione dei servizi di supporto di interesse generale, ovvero:

- la gestione dei sistemi online per l'accesso e la ricerca bibliografica tra le fonti cartacee ed elettroniche;
- la gestione del sito web del Sistema delle Biblioteche di Ateneo;
- la gestione delle risorse elettroniche;
- i servizi di prestito interbibliotecario e ricerche bibliografiche;
- la gestione di collezioni speciali: Fondo storico di Ateneo e Archivio tesi.

Il sistema integrato utilizzato è GEAC Advance versione 6.7, acquistato nel 1995 e integrato con il software Cyberpac per quanto riguarda l'OPAC. Il prodotto presenta diverse criticità, sia per quanto riguarda lo stadio di sviluppo, fermo da diversi anni¹⁷⁶, sia per quanto riguarda la gestione del sistema, riattivata all'inizio del 2006¹⁷⁷.

Attualmente la manutenzione del sistema è divisa tra SeGeBi e il Centro Servizi Informatici e di Telecomunicazioni di Ateneo (CSITA). Il SeGeBi si occupa di:

- creazione delle autorizzazioni e dei profili d'uso per l'accesso al sistema;
- configurazione dei moduli;
- modifiche globali ai dati di collocazione su set di record di esemplari;
- assistenza nella produzione di reportistica;

¹⁷⁶ Il sistema non presenta un'interfaccia grafica, né molte funzioni tipiche dei sistemi sviluppati nell'ultimo decennio (p.e. catalogazione derivata).

¹⁷⁷ Cfr. UNIVE-SBA [2009d]

- contatti con il servizio di assistenza remota fornito dalla ditta Infor (ex Geac France, Parigi);

Lo CSITA, invece, mantiene le funzioni di: presidio e manutenzione dei due server dedicati su cui sono installati GEAC Advance e Cyberpack, attività di backup¹⁷⁸.

Oltre al sistema integrato è stato avviato nel maggio 2007 il Progetto Multiricerca¹⁷⁹, che ha portato all'acquisto di MetaLib e SFX¹⁸⁰, due prodotti della Ex Libris per la ricerca federata e i link resolving.

La gestione del Sistema d'Ateneo è fortemente decentralizzata, per cui ciascuna biblioteca opera in maniera abbastanza indipendente dalle altre nello svolgimento delle proprie funzioni. Le uniche funzioni gestite a livello centralizzato dal SeGeBi riguardano principalmente: il prestito interbibliotecario, la gestione di fondi speciali, la gestione del materiale elettronico, la manutenzione del sistema integrato.

7.2. L'analisi dei flussi operativi automatizzabili

In questo contesto, i flussi operativi legati alla gestione bibliotecaria possono variare principalmente in base al tipo di biblioteca (di area, di dipartimento, di servizio didattico, ecc.). Ciononostante, l'utilizzo dello stesso sistema integrato per un così lungo tempo, rende possibile un'analisi generalizzata di tali flussi, fortemente determinati dalla condivisione del medesimo software.

È stata quindi scelta la Biblioteca di Area Economica come caso base per l'analisi dei flussi di lavoro a Ca' Foscari, limitatamente a quei processi legati a operazioni automatizzabili tramite l'utilizzo della versione 3.00.02 del sistema integrato Koha.

¹⁷⁸ Cfr. UNIVE-SBA [2009c].

¹⁷⁹ Cfr. UNIVE-SBA [2009b].

¹⁸⁰ MetaLib è installato su una piattaforma messa a disposizione dall'Università di Padova. Quest'ultima ospita anche la piattaforma per SFX, acquistato attraverso un contratto consortile anche dall'Università di Parma.

Per questo motivo si è scelto di concentrarsi sulle procedure relative a: acquisizioni, catalogazione, gestione periodici, circolazione¹⁸¹.

Per quanto riguarda i flussi relativi alla ricerca bibliografica, che coinvolgono utenti e bibliotecari, ci si limita a segnalare l'importanza di un catalogo online che, oltre alle funzioni implementate comunemente dagli OPAC¹⁸², fornisca la possibilità di attivare contenuto accresciuto per le notizie bibliografiche (p.e. copertine dei libri) oppure la creazione di contenuti da parte degli utenti (p.e. recensioni) in linea con gli sviluppi dei cataloghi di ultima generazione.

7.2.1. Metodo d'indagine

Il metodo utilizzato per studiare la possibilità di utilizzo di Koha ha previsto innanzitutto l'installazione di un ambiente di prova su una macchina virtuale messa a disposizione sui server del Dipartimento di Informatica e lo studio della documentazione ufficiale del progetto presente sul sito della comunità.

Successivamente sono state condotti una serie di colloqui con i vari bibliotecari competenti dei diversi processi. Il risultato di queste interviste è stato sintetizzato attraverso l'uso di Diagrammi Use Case basati sullo standard UML, in grado di descrivere l'insieme di funzionalità coinvolte in un certo processo.

Lo schema grafico, d'accompagnamento al testo, evidenzia:

- il sistema di riferimento, rappresentato da un rettangolo, contiene i diversi casi d'uso che partecipano al processo e che nel loro insieme descrivono i possibili utilizzi del sistema;
- il caso d'uso, rappresentato da un'ellisse, specifica una particolare funzionalità, traducibile in una serie di azioni che il sistema offre agli attori;

181 Non verranno quindi presi in considerazione i flussi operativi relativi a: prestito interbibliotecario, document delivery, accesso all'archivio tesi o fondi speciali, gestione risorse elettroniche, ecc.

182 Cfr. § 3.3.3. Modulo per il catalogo online

- l'attore, rappresentato da un omino stilizzato, specifica l'utilizzatore del sistema, il quale è esterno ad esso e ricopre dei ruoli funzionali nel processo; può essere un essere umano, un'organizzazione o un altro sistema (hardware o software);
- relazioni tra gli elementi, che possono essere:
 - associazione di comunicazione, ovvero l'interazione tra l'attore e il caso d'uso. Questa può essere asincrona e priva di direzione ed indica semplicemente l'esistenza di una comunicazione tra il sistema e l'attore.
 - generalizzazione/specializzazione, rappresentata da una linea continua con una freccia bianca, connette un caso d'uso generale ad uno o più casi d'uso specializzati, che ne ereditano tutte le caratteristiche;
 - include, rappresentato da una linea tratteggiata con freccia aperta, indica che un caso d'uso "base" prevede, o può prevedere, la chiamata di un caso d'uso "incluso" durante lo svolgimento del suo processo;
 - extend, rappresentato da una linea tratteggiata con freccia aperta, indica che il caso d'uso "base" comporta, o può comportare, l'utilizzo esteso delle caratteristiche del caso d'uso chiamato.

Al termine dell'analisi dei processi sono stati effettuati dei riscontri di natura tecnica tra le attuali procedure utilizzate e l'installazione di prova presso il Dipartimento di Informatica. L'assenza di dati bibliografici e amministrativi consistenti, da utilizzare per un'eventuale fase di test, ha limitato le possibili proposte di adeguamento a considerazioni generali circa le implementazioni minime per il funzionamento presso il Sistema Bibliotecario d'Ateneo.

7.2.2. Flusso operativo per le acquisizioni

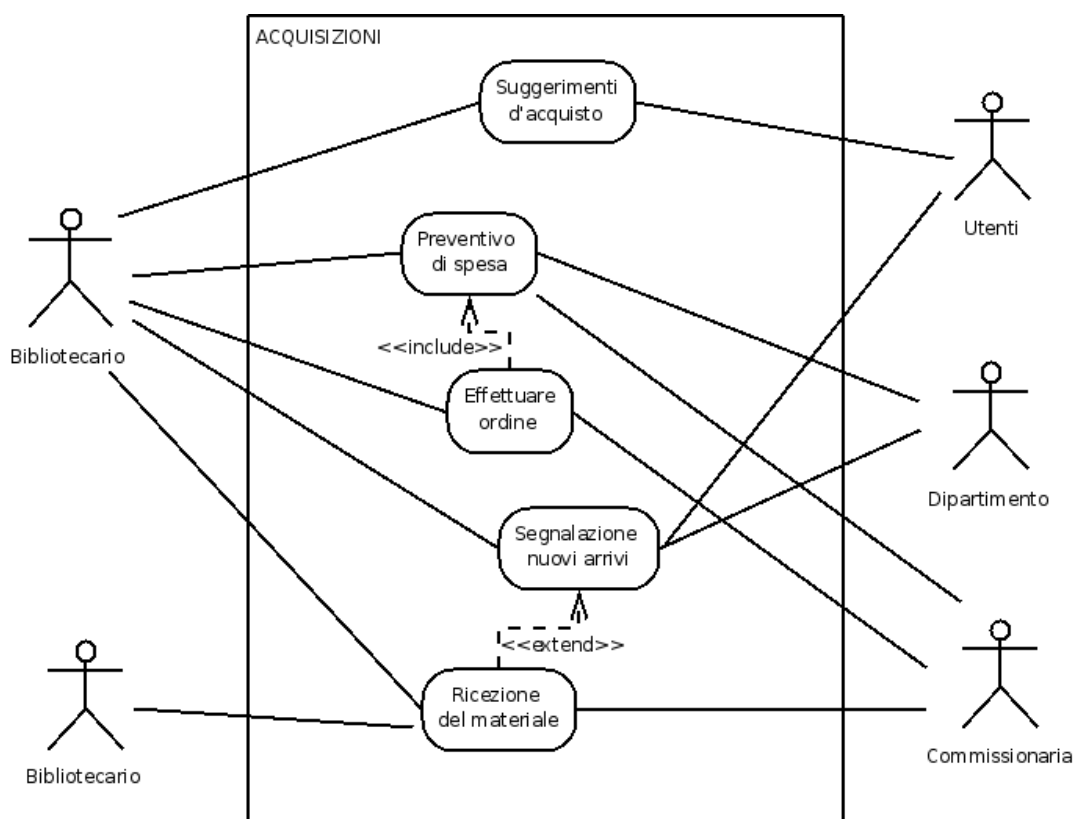


Figura 17: Workflow acquisizioni

Il suggerimento d'acquisto viene effettuato dagli utenti (studenti e docenti) compilando un apposito form online. Le richieste di acquisto vengono raccolte dal bibliotecario il quale verifica la correttezza bibliografica e l'eventuale presenza nelle collezioni della biblioteca.

Il bibliotecario, attraverso una o più librerie commissionarie, ottiene dei preventivi di spesa per l'acquisto; una volta scelta la commissionaria, viene effettuata una richiesta di approvazione di spesa da parte del docente responsabile dei fondi dedicati alla rispettiva area disciplinare. La segreteria amministrativa della BEC procede alla registrazione dell'impegno di spesa.

A questo punto il bibliotecario può effettuare l'ordine presso la commissionaria, questo comporta, oltre all'invio dell'ordine, la registrazione nel sistema di una

minima descrizione bibliografica, i dati amministrativi sul fornitore e il fondo utilizzato. I fondi possono appartenere alla biblioteca o fare parte di un'attività di ricerca presso un dipartimento.

La ricezione del materiale comporta la registrazione dell'arrivo del libro e della relativa fattura. In questo senso il rapporto con la commissionaria potrebbe essere legato anche all'invio di solleciti sul ritardo di arrivo del materiale o sull'emissione della fattura.

A questo punto oggetto e fattura passano nelle mani dell'addetto alla catalogazione. Una volta completata la catalogazione, viene trasmessa la fattura al dipartimento e viene segnalato l'arrivo dell'oggetto al richiedente. Nel caso di docenti o materiale acquistato tramite fondi di ricerca, la comunicazione avviene di solito tramite mail e il richiedente ha un certo tempo a disposizione, che può variare, per avere accesso esclusivo all'oggetto acquisito. In ogni caso l'addetto alle acquisizioni stila un Bollettino delle Nuove Acquisizioni, estraendo dal sistema le informazioni sugli ultimi arrivi, che viene pubblicato sul sito d'Ateneo.

All'interno del processo sono necessarie estrazione di dati dal sistema per quanto riguarda i contatti con gli utenti e i fornitori, nonché report sulla ricezione e registrazione di nuove acquisizioni.

7.2.3. Flusso operativo per la catalogazione

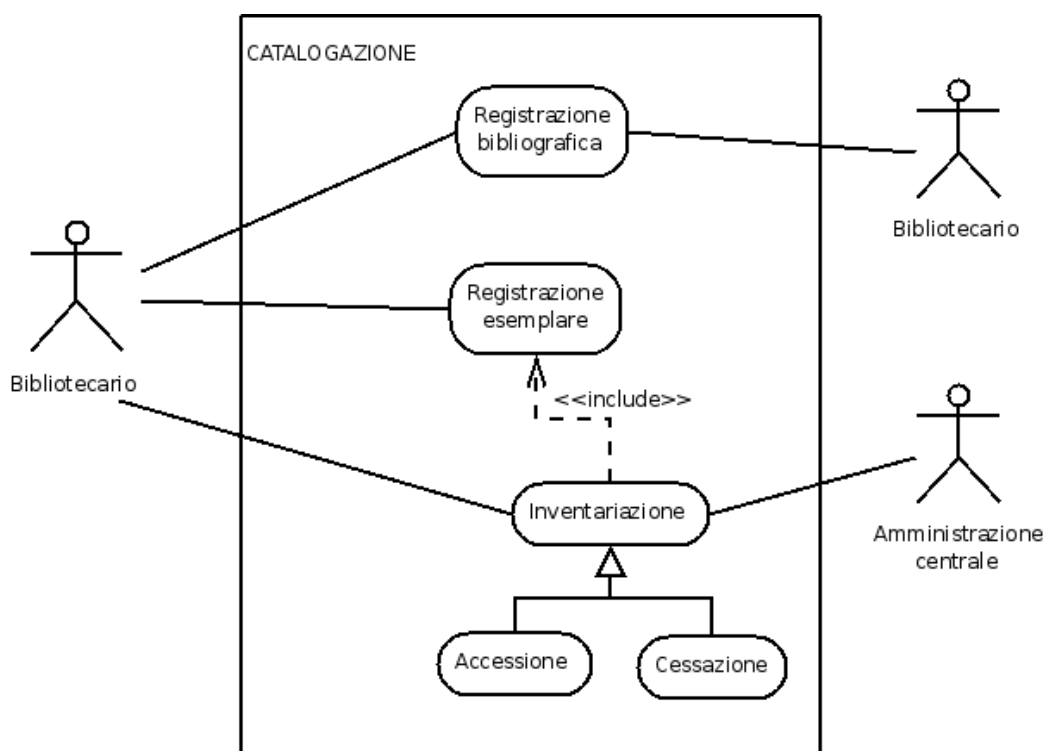


Figura 18: Workflow catalogazione

Il bibliotecario che si occupa della catalogazione riceve, dal processo di acquisizione, il materiale da catalogare e la fattura e procede all'arricchimento della registrazione bibliografica sintetica precedentemente fatta in sede di acquisizione.

Successivamente si occupa di registrare nel sistema le informazioni relative alla copia in possesso. Sempre durante la catalogazione, all'oggetto viene assegnato un numero di inventario relativo alla nuova acquisizione. Il processo di inventariazione prevede la tenuta del Registro Patrimoniale dei beni bibliografici dove vengono registrati: numero d'inventario, fattura, buono di carico, valore del bene. Copia del Registro viene trasmessa all'amministrazione centrale.

Il processo prevede il recupero di dati sia per quanto riguarda descrizioni bibliografiche a cui collegare nuove copie, sia per quanto riguarda i dati sulle copie per il Registro Patrimoniale.

7.2.4. Flusso operativo per la gestione dei periodici

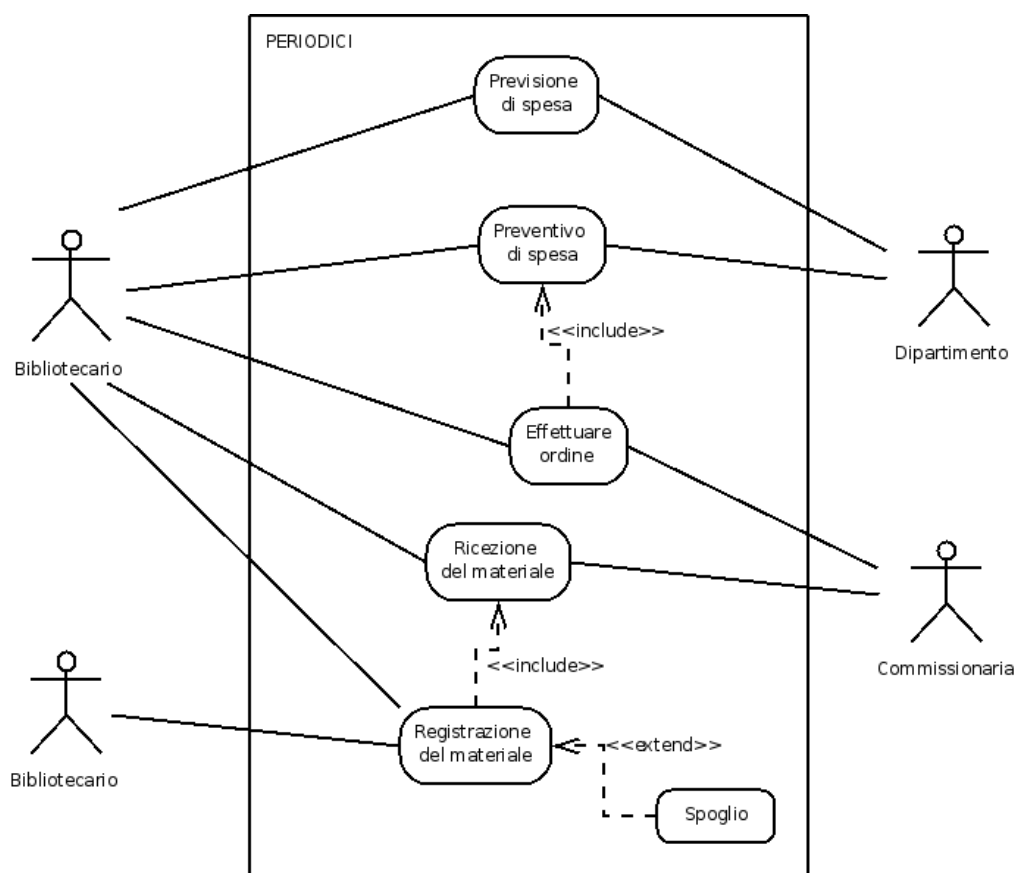


Figura 19: Workflow periodici

Dai dipartimenti vengono inviate le liste dei titoli in abbonamento da rinnovare. Il bibliotecario richiede quindi un preventivo che, se approvato, permette di procedere all'ordine.

La comunicazione relativa all'ordine tra il bibliotecario e la commissionaria può comprendere vari messaggi tra cui solleciti relativi ai ritardi nella consegna.

La ricezione del materiale prevede un controllo ed eventuali altre comunicazioni nel caso di problematiche riscontrate (p.e. fattura sbagliata).

La registrazione del materiale all'arrivo comporta la registrazione di dati amministrativi e l'inventariazione. In alcuni casi è previsto lo spoglio del periodico.

Nel caso di fornitura di materiale elettronico (p.e. accesso a basi dati online) interviene anche il system librarian che provvederà a fornire l'accesso ai materiali nell'apposito spazio sul sito d'ateneo.

All'interno di questa procedura il bibliotecario utilizza gli strumenti di estrazione dati principalmente per effettuare la previsione di spesa e monitorare lo stato degli ordini.

7.2.5. Flusso operativo per la circolazione

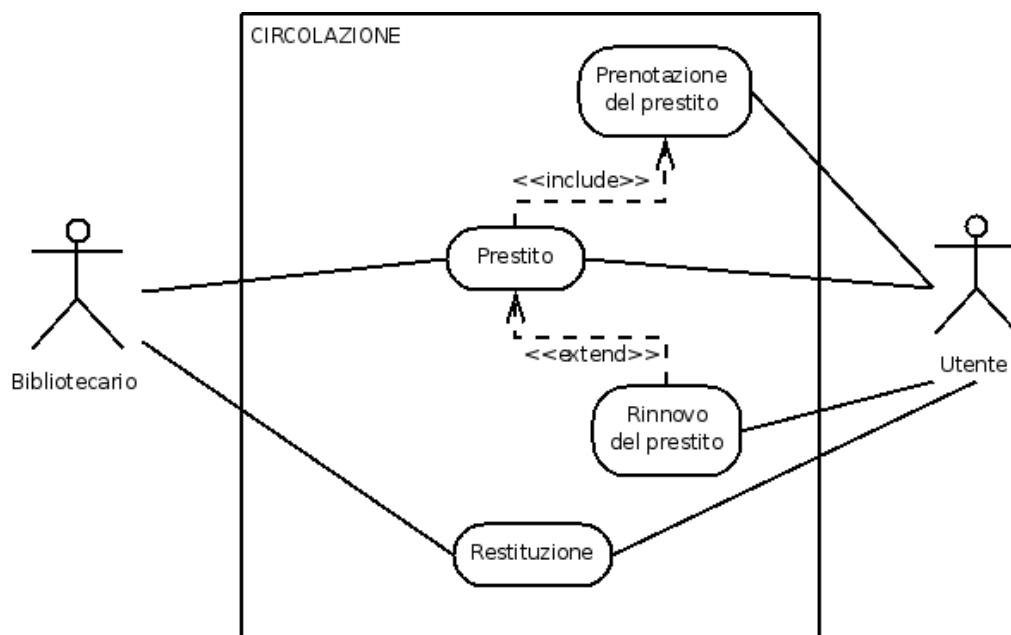


Figura 20: Workflow circolazione

L'addetto alla circolazione riceve una richiesta di prestito dall'utente, precedentemente registrato al sistema, e consegna la copia richiesta. Il prestito è sottoposto a regole definite dalla singola biblioteca.

La prenotazione può essere preceduta da una richiesta che comporta la creazione di una lista d'attesa sulle prenotazioni per le copie in prestito. Inoltre il prestito può

essere rinnovato a distanza, sempre secondo le regole per la circolazione definite dalla biblioteca.

La restituzione comporta la registrazione nel sistema da parte del bibliotecario della consegna del materiale. Inoltre la comunicazione con l'utente circa la restituzione potrebbe comportare eventuali solleciti o addirittura la sospensione dal prestito

Anche in questo caso è necessario da parte del bibliotecario l'utilizzo di strumenti di report per l'estrazione di dati sugli utenti registrati al sistema e sulle copie in circolazione.

7.3. Considerazioni per l'adeguamento del software

Comparando le funzionalità di Koha v. 3.0.2 con i vari processi messi in atto all'interno del sistema bibliotecario, si possono rilevare alcune criticità relative soprattutto ai singoli moduli.

Per quanto riguarda il modulo acquisizioni, si segnala la necessità di creare una soluzione relativa alla gestione inventariale. Una possibile soluzione alla tenuta del Registro Patrimoniale potrebbe essere la creazione di un'apposita tabella, collegata alle tabelle *items*, *aqbooksellers* e *aqorders* contenente: identificativo del buono di carico, data di registrazione, numero di fattura, identificativo del fornitore, numero d'inventario. A questo bisognerebbe associare poi la creazione di un'apposita interfaccia web e alcuni reports predefiniti per la stampa del registro e le estrazioni di dati utili suddivisi in base alla tipologia di materiale.

Per quanto riguarda il modulo per la catalogazione si segnalano alcune criticità rispetto alla gestione dei file d'autorità, non ancora particolarmente avanzata in Koha. Non sarebbe da sottovalutare inoltre la possibilità di arricchire il sistema di suggerimenti e avvisi per il catalogatore in fase di creazione della registrazione bibliografica.

Infine è da segnalare l'assenza del modulo per il collegamento all'Indice SBN, componente ritenuto molto utile all'interno del Sistema d'Ateneo ma il cui sviluppo non sembra essere previsto, data la ridotta comunità di interessati in Italia.

Oltre a queste criticità, la versione attuale di Koha è particolarmente adatta alla gestione di un Sistema Bibliotecario prevedendo una buona granularità in termini di permessi di accesso da parte dello staff e per quanto riguarda la gestione della circolazione in presenza di diverse biblioteche appartenenti al sistema. Gli strumenti per la reportistica, utilizzati frequentemente nella gestione di ogni processo, sono facilmente implementabili e, grazie al completo accesso alla base dati, personalizzabili da chiunque in possesso di una conoscenza minima del linguaggio SQL.

Inoltre per quanto riguarda la frequente necessità di scambio di messaggi con: utenti, fornitori, entità amministrative, organizzazioni o altre biblioteche, è prevista la possibilità di definire messaggi precompilati utilizzabili dai vari moduli, basati direttamente sui campi delle tabelle MySQL.

L'utilizzo di standard per la creazione del software lo rendono facilmente integrabile con altri sistemi, come RFID o sistemi per il self-check¹⁸³. A questo proposito è utile segnalare che la prossima release del software (3.2) dovrebbe comprendere l'integrazione con EDI, mentre è stata implementata la possibilità di autenticazione basata su CAS per l'OPAC già nella versione 3.00.05. È già presente invece la possibilità di integrazione con PazPar2, uno strumento per metaricerche sviluppato sempre dalla Index Data. È in via sperimentale anche l'integrazione con il protocollo OAI-PMH.

In questo senso è possibile riscontrare la forte attitudine all'interoperabilità di un software open source aderente agli standard e sviluppato a livello mondiale per automatizzare realtà anche molto differenti tra loro.

¹⁸³ Attualmente non implementati a Ca' Foscari.

7.4. Suggerimenti per un'ipotesi di migrazione

Anche se da un punto di vista prettamente tecnologico è possibile evidenziare una generale maturità del software per un'ipotesi di implementazione a livello di Sistema d'Ateneo, va segnalata la criticità relativa all'utilizzo in produzione da parte di una realtà di medie dimensione, per cui, attualmente, non esistono altri esempi in Italia.

L'attitudine sperimentale di un progetto di migrazione verso un sistema open source come Koha, potrebbe tradursi nella definizione di una serie di passaggi aggiuntivi rispetto ad un comune processo di migrazione verso un sistema proprietario.

Per quanto riguarda la realtà di Ca' Foscari, un possibile piano operativo potrebbe essere il seguente:

1. Istituzione di un comitato guida, ovvero un gruppo che abbia come obiettivo la definizione degli obiettivi e delle tempistiche del progetto, nonché ricopra un ruolo di monitoraggio e valutazione dei risultati.
2. Creazione di un gruppo di lavoro, non necessariamente interno all'Ateneo, preposto all'analisi dell'attuale flusso operativo del Sistema Bibliotecario, considerando sia le singole biblioteche che il sistema nel suo insieme, in grado di produrre la documentazione necessaria all'analisi delle singole procedure, sia in funzione di un'ottimizzazione del lavoro che della possibile automazione dei processi.
3. Creazione di diversi gruppi di lavoro tecnici, suddivisi per modulo (acquisizione, catalogazione, circolazione, OPAC, periodici) e formati da personale informatico e bibliotecari, in grado di definire le criticità del sistema e svilupparne i componenti mancanti.
4. Definizione di un protocollo di sviluppo da utilizzare per le modifiche del software; questo protocollo potrebbe definire anche la creazione di alcune figure preposte alla documentazione del progetto di migrazione e al

mantenimento dei rapporti con la comunità di utenti e sviluppatori di Koha.

5. Installazione e configurazione di un ambiente preposto all'analisi ed eventuale correzione dei dati estratti dal precedente sistema integrato; lo stesso ambiente potrebbe fornire la piattaforma di testing su cui installare la versione del software necessaria ai vari gruppi di lavoro per delinearne le criticità.
6. Formazione di gruppi di test da parte dei gruppi di lavoro tecnici, in grado di effettuare sessioni di prova su una prima installazione del sistema e di fornire i feedback necessari sia al reparto tecnico che al settore documentazione.
7. Sviluppo di script in grado di testare automaticamente il sistema in situazioni critiche.
8. Training del personale bibliotecario a cura dei gruppi di lavoro e di testing.
9. Installazione del sistema e passaggio in produzione.
10. Installazione di strumenti per la segnalazione di bug da parte di staff e utenti e correzione di eventuali problematiche.
11. Creazione di un gruppo di mantenimento del software, da affiancare al ruolo di system librarian, in grado di: mantenere i contatti con la comunità Koha, arricchire la documentazione dove necessario, collaborare alla traduzione di nuove componenti, fornire il feedback sul funzionamento del sistema e lo stimolo allo sviluppo di nuovi componenti.

Il piano operativo evidenzia da un lato la necessità di sinergia tra i diversi gruppi di lavoro, formati di volta in volta da personale informatico e bibliotecari, piuttosto che da utenti del sistema, dall'altro pone la questione dello sviluppo del software come condizione necessaria alla sua implementazione.

8. Conclusioni

Al termine di un lavoro di indagine su un sistema integrato open source e la sua possibile applicazione ad un Sistema Bibliotecario d'Ateneo come quello di Ca' Foscari è possibile effettuare diverse considerazioni, sia per quanto riguarda il proseguimento di tale indagine, sia, più in generale, circa l'adozione di sistemi open source da parte delle biblioteche italiane.

Per quanto riguarda un concreto studio di fattibilità sulla possibile applicazione di Koha al Sistema Bibliotecario si segnala la necessità di proseguire l'indagine in due direzioni.

Da un lato una più approfondita analisi del Sistema nel suo complesso, che comprenda tutti quegli attori e quelle procedure che rendono possibile considerarlo tale ed evidenziando tutte le criticità relative alla sua attuale organizzazione. Dall'altro sarebbe necessario predisporre un ambiente di test in grado di simulare il funzionamento del sistema in produzione. In questo senso, con ambiente di test non si dovrebbe intendere esclusivamente l'installazione del software su un server e la corretta configurazione secondo i parametri in uso dall'Ateneo, ma andrebbe considerata anche la creazione di un gruppo di lavoro in grado di coinvolgere i diversi partecipanti al sistema integrato, ovvero informatici, bibliotecari e utenti.

Proprio la trasversalità di un tale gruppo evidenzia le due principali criticità, non solo del Sistema d'Ateneo, ma probabilmente di molti sistemi bibliotecari italiani: la

manca di system librarian e lo scarso coinvolgimento degli utenti nelle scelte di selezione del sistema integrato

Il system librarian dovrebbe rappresentare la figura di riferimento del processo di automazione, coniugando le competenze dell'informatico e del bibliotecario o, se non altro, fungendo da mediatore tra le due professionalità. Il ruolo, tanto diffuso nel mondo anglosassone quanto scarsamente riconosciuto in Italia¹⁸⁴, risponde alla domanda circa le competenze necessarie a guidare la scelta e la gestione di un sistema integrato.

Il maggiore coinvolgimento degli utenti nella selezione del sistema integrato vorrebbe essere invece un suggerimento per migliorare un servizio che proprio all'utente è destinato. Se è vero che sta cambiando il rapporto tra utente e biblioteche¹⁸⁵, non sarebbe così assurdo pensare di sfruttare le potenzialità del confronto con gli utenti, ormai abituati dal web 2.0 a fornire i propri contributi per la personalizzazione di un servizio¹⁸⁶.

Per quanto riguarda l'adozione di sistemi integrati open source in biblioteca, le considerazioni tecnologiche restano ancora legate ad altre di tipo etico-filosofico, rendendo complesso rispondere alla domanda sul perché scegliere software open source.

Dal punto di vista tecnologico, l'ampia diffusione e la longevità del progetto Koha, così come i risultati ottenuti dal progetto Evergreen, confermano la concreta possibilità di considerare i sistemi integrati open source sufficientemente maturi ed equiparabili a molte alternative proprietarie.

Certamente non sono da sottovalutare gli svantaggi della scelta open source in Italia: carenza di supporto tecnico e carenza di case studies. È però opportuno considerare che, paradossalmente, sono gli stessi svantaggi ad alimentarsi in maniera ridondante, scoraggiando le sperimentazioni da parte delle biblioteche, fortemente preoccupate di ottenere garanzie di stabilità sul lungo termine che solo

184 Cfr. Spinello [2007].

185 Cfr. Marchitelli [2008a].

186 Cfr. Ryan [2004]

un rivenditore commerciale sembrerebbe offrire.

Esistono tuttavia diverse situazioni in cui queste garanzie si dimostrano infondate, e non si fa riferimento esclusivamente al turbolento mercato anglosassone, fatto di fusioni, fallimenti e cambi nelle strategie di sviluppo del gruppo aziendale¹⁸⁷, ma anche ad alcune situazioni molto vicine, come per esempio la scelta di utilizzo di un sistema proprietario nello sviluppo di SBN e i ritardi nello sviluppo che questo può avere causato¹⁸⁸, così come i disagi che ha provocato a Ca' Foscari la fine dello sviluppo di GEAC, il sistema integrato in uso¹⁸⁹. Anche se molti rivenditori di sistemi integrati da diversi anni hanno abbracciato l'uso di standard per lo sviluppo dei loro prodotti, viene da chiedersi se la strategia più lungimirante non sia quella di mantenere il pieno controllo su dati e codice del sistema scelto, così come permette l'adozione di sistemi open source.

Sempre per quanto riguarda l'aspetto tecnico e più precisamente i timori circa le difficoltà d'installazione, che si considera legata a solide competenze informatiche, andrebbe considerato come molti sistemi, basandosi sull'architettura LAMP, permettano l'operazione in maniera relativamente semplice anche a personale non specializzato¹⁹⁰. Senza considerare come il libero accesso al software potrebbe ulteriormente incentivare sperimentazioni a costo zero in questo senso.

Le considerazioni etico-filosofiche riguardano invece l'insieme di valori promossi dal movimento per il software libero e le affinità con i valori che contraddistinguono il mondo delle biblioteche.

Innanzitutto il medesimo approccio alla cooperazione, di primaria importanza in qualsiasi processo di sviluppo tanto per le biblioteche, quanto per il software open source. Basterebbe paragonare, ad esempio, i sistemi di catalogazione partecipata ai sistemi di controllo di versione utilizzati per lo sviluppo collaborativo di software.

Oppure anche l'attenzione e la partecipazione alla comunità di riferimento, che per

187 Cfr. § 3.4. La dis-integrazione dei sistemi integrati.

188 Cfr. § 5.1. La lunga via della cooperazione.

189 Alla fine del 2009 sono state avviate le procedure per il passaggio ad un novo sistema integrato.

190 Cfr. Cargile [2005].

le biblioteche, così come per i progetti open source, è rappresentata innanzitutto dagli utenti del sistema.

Infine va considerato uno dei valori più importanti, ovvero la promozione del libero accesso al sapere, inteso come servizio all'umanità¹⁹¹. In questo senso biblioteche e open source condividerebbero la stessa "cultura del dono"¹⁹².

E *koha*, in maori, significa proprio "dono".

191 Cfr. Dione – Savard [2008] e il progetto eIFL-FOSS per alcune testimonianze dell'importanza dell'open source nel contrasto al digital divide nei Paesi in via di sviluppo.

192 Cfr. Morgan [2004c].

Appendice A

Guida di installazione di Koha su Debiana Lenny con MySQL 5

=====

Feedback/bug reports: Koha Developer's List:
<http://lists.koha.org/mailman/listinfo/koha-devel>

Ultima modifica a questo documento: 13 Marzo 2009

Istruzioni di installazione

=====

Tutti i comandi possono essere eseguiti come utente normale con privilegi sudo, come indicato, oppure eseguendoli direttamente come root.

1. Preparare il sistema e installare le dipendenze

1.1 Installare Debian Lenny via Businesscard CD

Vedi <http://www.debian.org/releases/lenny/debian-installer/>

1.2 Imposta la localizzazione

Il tuo character set dovrebbe essere impostato su UTF-8, così come dovrebbero essere impostati così anche Apache2 e MySQL 5. Questo passaggio è MOLTO IMPORTANTE per sistemi conformi a UNICODE. Per approfondire leggi il seguente documento con attenzione:

<http://wiki.koha.org/doku.php?id=encodingscratchpad>

IMPORTANTE: Tutti i passaggi elencati di seguito per Apache2, MySQL 5, ecc. DEVONO essere seguiti PRIMA di installare Koha.

1.3 Imposta i sorgenti apt per i pacchetti Git, Yaz e Zebra

Modifica il tuo file `/etc/apt/sources.list` aggiungendo le seguenti linee:

```
# Index Data
deb http://ftp.indexdata.dk/debian lenny main
deb-src http://ftp.indexdata.dk/debian lenny main
```

I pacchetti Index Data sono segnati con una chiave che puoi installare come segue:

```
$ wget http://ftp.indexdata.dk/debian/indexdata.asc
$ sudo apt-key add indexdata.asc
```

1.4 Ottenere Koha

1.4.1 Opzione A: scaricare Koha tramite Git (opzionale)

```
$ sudo apt-get install git-core git-email
$ git clone git://git.koha.org/pub/scm/koha.git kohaclone
$ cd kohaclone
$ git checkout -b myinstall origin
```

Nota: per maggiori informazioni su Git, per favore leggi la Koha Git Usage Guide:

http://wiki.koha.org/doku.php?id=en:development:git_usage

1.4.2 Opzione B: scaricare Koha da <http://download.koha.org>

Trova l'ultima versione stabile di Koha su <http://download.koha.org> e ottienilo con

```
$ wget http://download.koha.org/koha-3.00.01.tar.gz
```

1.5 Installare dipendenze Debian aggiuntive

IMPORTANTE: Per risolvere le dipendenze Perl NON incluse nel pacchetto ufficiale dovresti usare solo CPAN. Sei avvisato!

Usando il file `debian-lenny.packages` incluso nella cartella di Koha, eseguite il seguente comando:

```
$ sudo dpkg --set-selections < install_misc/debian-lenny.packages
```

Adesso avviate `dselect`:

```
$ sudo apt-get install dselect
$ sudo dselect
```

Scegliere `[I]ninstall` e accettare i pacchetti da installare (premendo Invio)

(potrebbe volerci un po' di tempo)

Scegliere [C]onfigure, [R]emove e [Q]uit fino alla conclusione di dselect.

1.6 Installare le dipendenze Perl non comprese nei pacchetti sorgenti di Debian Lenny

Eseguire il seguente comando:

```
$ sudo cpan Algorithm::CheckDigits SMS::Send HTTP::OAI
```

Nota: potresti dover eseguire l'inizializzazione di CPAN se non l'hai mai eseguito prima:

```
/etc/perl/CPAN/Config.pm initialized.
```

CPAN is the world-wide archive of perl resources. It consists of about 100 sites that all replicate the same contents all around the globe. Many countries have at least one CPAN site already. The resources found on CPAN are easily accessible with the CPAN.pm module. If you want to use CPAN.pm, you have to configure it properly.

If you do not want to enter a dialog now, you can answer 'no' to this question and I'll try to autoconfigure. (Note: you can revisit this dialog anytime later by typing 'o conf init' at the cpan prompt.)

Are you ready for manual configuration? [yes]

Una volta completata la configurazione, CPAN installerà i moduli Perl.

2. Configurazione delle dipendenze

2.1 Aggiorna la password MySQL del root (se non l'ha già fatto dselect)

```
$ sudo mysqladmin password <password>
```

2.2 Crea il database per Koha

Crea il database e l'utente con i privilegi associati:

```
$ mysqladmin -uroot -p<password> create <nomekohadatabase>
$ mysql -uroot -p<password>
```

Welcome to the MySQL monitor. Commands end with ; or \g.

Your MySQL connection id is 22
Server version: 5.0.51a-24 (Debian)

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

```
mysql> grant all on <nomekohadatabase>.* to '<utentekohadatabase>'@'localhost'  
identified by '<passwordutentekohadatabase>';  
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> flush privileges;  
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> quit
```

2.3 Testa il Parser SAX e correggi se necessario

Devi essere sicuro che stai usando il parser SAX XML::LibXML, non Expat o Pureperl, coi quali sono stati riscontrati bug coi caratteri pre-composti. Puoi testare il parser SAX eseguendo:

```
$ cd koha  
$ misc/sax_parser_print.pl
```

Dovresti leggere qualcosa di simile a:

```
XML::LibXML::SAX::Parser=HASH(0x81fe220)
```

Se stai usando PurePerl o Expat, dovrai cambiare il tuo file ini, solitamente situato in:

```
/etc/perl/XML/SAX/ParserDetails.ini
```

3. Eseguire l'installer di Koha

```
$ perl Makefile.PL  
( rispondi alle domande )  
$ make  
$ make test  
$ sudo make install
```

4. Configurare e avviare Apache

```
$ sudo ln -s /etc/koha/koha-httpd.conf /etc/apache2/sites-available/koha  
(ricorda che il percorso a koha-httpd.conf potrebbe essere diverso in  
base alle tue scelte di installazione)
```


Aggiungi le seguenti righe a /etc/apache2/ports.conf:

```
Listen 80
Listen 8080
```

Esegui i seguenti comandi:

```
$ sudo a2enmod rewrite
$ sudo a2ensite koha
$ sudo apache2ctl restart
```

5. Configurare e avviare Zebra

Nota: si raccomanda di demonizzare il processo di Zebra e aggiungerlo al tuo profilo di avvio. Per una installazione di test o di sviluppo, potrebbe essere utile eseguire Zebra da linea di comando. Scegli tra una delle due opzioni suggerite di seguito, o crea la tua :-)

Nota: si raccomanda inoltre di creare un utente specifico per Koha, che dovresti aver specificato durante il processo di installazione. In alternativa, Zebra può essere configurato per essere eseguito dall'utente root.

Opzione 1: eseguire i processi di Zebra da linea di comando:

Zebra Server:
\$ sudo -u \${KOHA_USER} zebrasrv -f /etc/koha/koha-conf.xml
(ricorda che il percorso a koha-conf.xml potrebbe essere diverso in base alle tue scelte di installazione)

Nota: l'utente con cui hai eseguito Zebra sarà il solo utente con permessi di scrittura nell'indice Zebra; in modalità sviluppatore, potresti voler utilizzare l'utente di sistema.

Zebraqueue Daemon:
\$ sudo -u \${KOHA_USER} misc/bin/zebraqueue_daemon.pl

Nota: se lo esegui in questo modo, potresti voler rimandare l'avvio di zebraqueue finché non avrai eseguito il web installer (vedi punto 6).

Opzione 2: eseguire i processi di Zebra come demoni e aggiungerli all'avvio:

Ricorda che facendo riferimento a \$SCRIPT_DIR si intende la directory dove sono installati gli script di Koha, p.e. /usr/share/koha/bin.

Zebra Server:

```
$ sudo ln -s ${SCRIPT_DIR}/koha-zebra-ctl.sh /etc/init.d/koha-zebra-daemon
```

(Nota: \${SCRIPT_DIR} è di default /usr/share/koha/bin/ in una installazione standard)

```
$ sudo update-rc.d koha-zebra-daemon defaults
```

(Nota: vedi man chkconfig(8) su altre distro)

```
$ sudo ${SCRIPT_DIR}/koha-zebra-ctl.sh start
```

Zebraqueue Daemon:

```
$ sudo ln -s ${SCRIPT_DIR}/koha-zebraqueue-ctl.sh /etc/init.d/koha-zebraqueue-daemon
```

```
$ sudo update-rc.d koha-zebraqueue-daemon defaults
```

(Nota: vedi man chkconfig(8) su altre distro)

```
$ sudo ${SCRIPT_DIR}/koha-zebraqueue-ctl.sh start
```

6. Eseguire il Web Installer, popolare il database, prima configurazione dei settaggi

Punta il browser a <http://<servername>:8080/>

Dovrebbe reindirigerti al Web Installer dove potrai continuare il setup. Attraverso il Web Installer puoi installare dei dati di esempio per le biblioteche, gli utenti, ecc.

7. E poi?

Una volta terminata l'installazione, puoi importare e indicizzare record MARC anche da linea di comando (Nota: per importare in modalità batch i record MARC, puoi anche usare 'Prepara i records MARC per l'importazione' dall'area Strumenti del Client per lo staff Koha):

```
$ export KOHA_CONF=/usr/share/koha/etc/koha-conf.xml
```

(Nota: usa il percorso del tuo koha-conf.xml)

7.0 Programmare crontab jobs

Vuoi che Koha:

- spedisca mail?
- indicizzi automaticamente i nuovi records?
- generi notifiche avanzate o sui ritardi?
- segna gli items troppo in ritardo come persi?
- faccia una stima delle multe?

Queste e altre operazioni in background pianificate con regolarità sono gestite da crontab. Hai solo bisogno di esaminare il crontab file, modificarne una copia a tua piacimento e pianificarla.

```
$ crontab -l      # controlla semplicemente se ci sono altre operazioni pianificate
$ cp ./misc/cronjobs/crontab.example my_crontab
$ vi my_crontab   # seleziona le operazioni o aggiusta gli orari.
$                # Se avessi già delle righe pianificate, aggiungile in fondo.
$ crontab my_crontab # Questo sovrascrive i processi pianificati con quelli nuovi.
```

Se vuoi fare un check per sicurezza, puoi eseguire ancora `crontab -l`

7.1 Importare:

```
Dati bibliografici in formato MARC21
$ misc/migration-tools/bulkmarcimport.pl -file /path/to/marc.iso2709
Dati d'autorità in formato MARC21
$ misc/migration-tools/bulkauthimport.pl -file /path/to/auth.iso2709
```

7.2 Fast Index:

```
$ misc/migration-tools/rebuild_zebra.pl -b -w
```

Una volta terminata l'indicizzazione, sarà possibile cercare records nel tuo sistema. NOTA: se non esegui l'utility Fast Index, e hai programmato gli aggiornamenti degli indici come da punto 7.3, il processo di indicizzazione sarà eseguito in background, ma sarà tremendamente più lento.

7.3 Programma aggiornamenti regolari degli indici

esegui `rebuild_zebra.pl -b -a -z` come un regolare cron job, controlla `misc/cronjobs/crontab.example` per esempi di utilizzo

7.4 Per abilitare i servers pubblici Z39.50/SRU, avrai bisogno di editare `koha-conf.xml` e cambiare le opzioni `<listen>` per abilitare una porta TCP; poi riavvia il demone zebra.

UPGRADE

=====

Se stai aggiornando una versione precedente a Koha 3, puoi usare i seguenti comandi:

```
perl Makefile.PL --prev-install-log /path/to/koha-install-log
make
make test
sudo make upgrade
```

Istruzioni per la rimozione

=====

1) Fermare i servizi:

```
$ sudo a2dissite koha
$ sudo rm /etc/apache2/sites-available/koha
$ sudo apache2ctl restart
```

```
$ sudo update-rc.d koha-zebra-daemon remove
$ sudo rm /etc/init.d/koha-zebra-daemon
```

2) Rimuovere database e indici

```
# MySQL
$ mysql -u<kohauser> -p<kohapassword>
> drop database koha;

# Zebra Indexes
$ zebraidx -c <prefix>/etc/zebradb/zebra-biblios.cfg -g iso2709 -d biblios init
$ zebraidx -c <prefix>/etc/zebradb/zebra-authorities.cfg -g iso2709 -d authorities
init
```

3) Rimuovere le directories di installazione Koha e i files di configurazione

Non dimenticare di tutte le crontab entries.

Testato sul seguente sistema operativo

=====

- Debian Lenny 5.0

Altre note

=====

Questo file è parte di Koha

Koha è un software libero; puoi redistribuirlo e/o modificarlo entro i termini della GNU General Public License come pubblicata dalla Free Software Foundation; sia la versione 2 della licenza, che (come preferisci) ogni altra versione.

Koha è distribuito nella speranza che risulti utile, ma SENZA NESSUNA GARANZIA; nemmeno con la garanzia implicita di COMMERCIALIZZABILITA' o IDONEITA' PER UNO SCOPO SPECIFICO. Vedi la GNU General Public License per maggiori dettagli.

Insieme a Koha avresti dovuto ricevere anche una copia della GNU General Public License; altrimenti, scrivi alla Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA.

Bibliografia^{*}

Abram, Stephen [2009] *Integrated Library System Platforms on Open Source*, SirsiDynix position paper, Set. 2009.

Adamson, Veronica - Bacsich, Paul - Chad, Ken - Kay, David - Plenderleith, Jane [2008] *Library Management Systems Horizon Scan*, JISC & SCONUL, <<http://www.jisc.ac.uk/whatwedo/programmes/resourcediscovery/libraryms.aspx>>.

AIB-GBD [2005] *Manifesto per le biblioteche digitali*, a cura del Gruppo di studio sulle biblioteche digitali dell'AIB, in "AIB-WEB", Dic. 2005, <<http://www.aib.it/aib/cg/gbdigd05a.htm3>>.

AIB-SW [2008] *Software per OPAC*, repertorio creato e mantenuto dall'AIB, ultimo aggiornamento 12 Nov. 2008, <<http://www.aib.it/aib/opac/sw.htm>>.

Bahr, Ellen [2007] *Dreaming of a Better ILS*, "Computers in Libraries", 27 (Oct. 2007), n. 9, pp. 10-14.

Balas, Janet L. [2007] *Will the ILS Soon Be as Obsolete as the Card Catalog?*, "Computer in Libraries", 27 (Oct. 2007), n. 9, pp. 41-43.

Baldacci, Maria Bruna [1983] *Informatica e biblioteche: automazione dei sistemi informativi bibliotecari*, Roma, La Nuova Italia Scientifica, 1983.

Baldi, Paolo [2005] *Il bibliotecario e l'open source*, "Bibelot", 11 (2005), n. 2, <<http://www.aib.it/aib/sezioni/toscana/bibelot/0502/b0502b.htm>>

Barreau, Deborah [2001] *The hidden costs of implementing and maintaining information systems*, "The Bottom Line: Managing Library Finances", 14 (2001), n. 4, pp. 207-213.

Benvenuti, Nicola [2003] *L'importanza dei metadati nella costruzione della rete globale*, "Bollettino AIB", 43 (2003), n. 3, pp. 331-350.

* Gli indirizzi URL riportati sono stati tutti controllati l'ultima volta in data 5 marzo 2010.

Bertini Vanni [2002] *I sistemi di automazione in Italia*, in *Rapporto sulle biblioteche italiane 2001*, a cura di Vittorio Ponzani, "Bollettino AIB", 42 (2002), n. 4, pp. 475-500, <<http://www.aib.it/aib/boll/rapp01.htm>>.

Bertini, Vanni [2003] *Sistemi di automazione*, in *Rapporto sulle biblioteche italiane 2002*, a cura di Vittorio Ponzani, "AIB-WEB", 13 Ott. 2003, <<http://www.aib.it/aib/editoria/rbib/rbib02.pdf>>.

Bertini, Vanni [2009] *I sistemi di automazione in Italia*, in *Rapporto sulle biblioteche italiane 2007-2008*, a cura di Vittorio Ponzani; direzione scientifica di Giovanni Solimine, Roma, AIB, 2009, pp. 115-127.

Bisson, Casey [2007] *Open Source Software For Libraries*, con la collaborazione di Jessamyn West e Ryan Eby, "Library Technology Reports", 43 (May/Jun. 2007), n. 3, <<http://maisonbisson.com/oss4lib/>>.

Bollini, Andrea – Mornati, Susanna [2009] *L'open source strumento di efficienza. L'esperienza del modulo SUR+OA: resoconto di un anno di attività con la community di Dspace*, "Bollettino del CILEA", 114 (Dic. 2009), pp. 16-18, <<http://bollettino.cilea.it/include/getdoc.php?id=2181&article=787&mode=pdf>>.

Bonfield, Brett [2009], *W-E-B-S-I-T-E, Find Out What It Means To Me*, "In the Library with the Lead Pipe", 22 Jul. 2009, <<http://www.inthelibrarywiththeleadpipe.org/2009/w-e-b-s-i-t-e-find-out-what-it-means-to-me/>>.

Borgman, Christine L. [1997] *From acting locally to thinking globally: A brief history of library automation*, "Library Quarterly", 67 (1997), n. 3, pp. 215-249.

Boss, Richard W. [2008] *"Open Source" Integrated Library System Software*, "PLA Tech Notes", 14 Dec. 2008, <<http://www.ala.org/ala/mgrps/divs/pla/plapublications/platechnotes/opensourcesoftware.pdf>>.

Breeding, Marshall [2004] *The Many Facets of Managing Electronic Resources*, "Computers in Libraries", 24 (Jan. 2004), n. 1, pp. 25-28.

Breeding, Marshall [2005] *Re-Integrating the integrated library system*, "Computers in Libraries", 25 (Jan. 2005), n. 1, pp. 28-31 <<http://www.librarytechnology.org/ltg-displaytext.pl?RC=11340>>.

Breeding, Marshall [2007a] *An Industry Redefined: Private equity moves into the ILS, and open source support emerges*, "Library Journal", Apr. 2007, <<http://www.libraryjournal.com/article/CA6429251.html>>.

Breeding, Marshall [2007b] *The ILS: The Past, Present & Future*, presentazione al "Symposium on the Future of Integrated Library Systems", 13-15 Sep. 2007, Champaign, <<http://www.lincolntail.info/ilssymposium2007/agendapage.html>>.

Breeding, Marshall [2007c] *History of Library Automation*, grafico pubblicato in "Library Technology Guidelines", ultimo aggiornamento 2007, <<http://www.librarytechnology.org/automationhistory.pl>>.

Breeding, Marshall [2009a] *Automation system marketplace 2009: Investing in the Future*, "Library Journal", Apr. 2009, <<http://www.libraryjournal.com/article/CA6645868.html>>.

Breeding, Marshall [2009b] *Moving forward Through Tech Cycles*, "Computers in Libraries", 29 (May 2009), n. 5, pp. 19-21.

Breeding, Marshall [2009c] *Clifford Lynch clarifies views relative to open source ILS efforts*, "Library Technology Guides", 30 Oct. 2009, <<http://www.librarytechnology.org/blog.pl?ThreadID=134&BlogID=1>>.

Breeding, Marshall [2010] *Perceptions 2009: An International Survey of Library Automation*, "Library Technology Guides", 22 Jan. 2010, <<http://www.librarytechnology.org/perceptions2009.pl>>.

Bretthauer, David [2002], *Open Source Software: A History*, "Information Technology and Libraries", 21 (Mar. 2002), n. 1, pp. 3-10, <<http://www.ala.org/ala/mgrps/divs/lita/ital/2101bretthauer.cfm>>.

Bucchioni, Cinzia [2002] *SFX e OpenURL: gli esperimenti del team di Van de Sompel*, "Bibliotime", anno V (luglio 2002), n. 2, <<http://didattica.spbo.unibo.it/bibliotime/num-v-2/bucchion.htm>>.

Bucchioni, Cinzia – Spinelli, Serafina [2007] *Teoria e pratica del linking citazionale*, "Biblioteche oggi", 25 (2007), n. 8, pp. 70-86, <<http://www.bibliotecheoggi.it/2007/20070807001.pdf>>.

Byrum, John D. Jr. [2005] *Raccomandazioni per miglioramenti urgenti nell'Opac*, traduzione di Grazia Di Bartolomeo, "Biblioteche oggi", 23 (2005), n. 10, pp. 5-14.

Carassiti, Alessandra – Tajoli, Zeno [2009] *Koha: un software gestionale open source per le biblioteche*, "Bollettino del CILEA", 114 (Dic. 2009), pp. 30-33, <<http://bollettino.cilea.it/include/getdoc.php?id=2185&article=784&mode=pdf>>.

Cargile, Cathleen [2005] *Open source ILS for the non-systems librarian: a reality?*, "PNLA Quarterly", 69 (2005), n. 3, pp. 16-22.

Casey, Michael [2005] *Working towards a definition of Library 2.0*, in Library Crunch, 21 Oct., 2005,
<http://www.librarycrunch.com/2005/10/working_towards_a_definition_o.html>.

Cassella, Maria – Gargiulo, Paola [2006] *Il workflow delle risorse elettroniche remote*, “Biblioteche oggi”, 24 (2006), n. 6, pp. 46-58,
<<http://www.bibliotecheoggi.it/2006/20060604601.pdf>>.

Cassella, Maria [2007] *Le risorse elettroniche nei portali di biblioteche*, “Biblioteche Oggi”, 25 (2007), n.10, pp. 5-15,
<<http://www.bibliotecheoggi.it/content/20071000501.pdf>>.

Cervone, Frank [2007] *ILS Migration in the 21st Century*, “Computers In Libraries”, 27 (Jul./Aug. 2007), n. 7, pp. 6-8 e 60-62.

Chudnov, Daniel [2009] *What Librarians Still Don't Know About Free Software*, “Computers in Libraries”, 29 (Feb. 2009), n. 2, pp. 22-24.

Code4Lib [2009] *SirsiDynix: Integrated Library System Platforms on Open Source*, wiki page in Code4Lib Wiki, ultimo aggiornamento 22 Nov. 2009,
<http://wiki.code4lib.org/index.php/SirsiDynix:_Integrated_Library_System_Platforms_on_Open_Source>.

Colford, Scot [2009] *Explaining Free and Open Source Software*, “Bulletin of the American Society for Information Science and Technology”, 35 (Dec./Jan. 2009), n. 2, pp. 10-14.

Corrado, Edward M. [2009] *Are OSS ILSs redundant? Carl Grant and the SirsiDynix kerfuffle*, in *blog.ecorrado.us*, 08 Nov. 2009, <<http://blog.ecorrado.us/2009/11/08/are-oss-ilss-redundant-carl-grant-and-the-sirsidynix-kerfuffle/>>

Corrado, Edward M. [2010] *Library Perceptions Gossip*, in *blog.ecorrado.us*, 26 Jan. 2010, <<http://blog.ecorrado.us/2010/01/26/library-perceptions-gossip/>>.

Courant, Paul N. - Griffiths, Rebecca J. [2006] *Software and Collaboration in Higher Education: A Study of Open Source Software*, in ITHAKA Publication, 2006,
<http://www.ithaka.org/strategic-services/oss/OOSS_Report_FINAL.pdf>.

Cover [2005] *XML Applications and Initiatives*, “Cover Pages – online resource for markup language technologies”, Robin Cover editor, ultimo aggiornamento 25 Giu. 2005, <<http://xml.coverpages.org/xmlApplications.html>>.

Coyle, Karen [2010] *Understanding the Semantic Web: Bibliographic Data and Metadata*, “Library Technology Reports”, 46 (2010), n. 1, pp. 5-13.

Crocetti, Luigi [1986] *Relazione introduttiva*, in *La cooperazione: il Servizio Bibliotecario Nazionale: atti del 30. Congresso nazionale della Associazione Italiana biblioteche*, Giardini Naxos, 21-24 novembre 1982, a cura di Antonella Aquilina D'Amore, Università di Messina, Facoltà di lettere e filosofia, Centro studi umanistici, Messina, 1986, p. 31-45.

Danesi, Daniele – Giaccai, Susanna [1986] *Un programma per le piccole biblioteche: il sistema Tinlib: descrizione e istruzioni per l'uso*, "Biblioteche Oggi", 4 (1986), n. 2, pp. 37-42.

Dempsey, Lorcan [2005a] *The integrated library system that isn't*, in *Lorcan Dempsey's weblog: on libraries, services and networks*, 22 Feb. 2005, <<http://orweblog.oclc.org/archives/001378.html>>.

Dempsey, Lorcan [2005b] *Discover, locate, ... vertical and horizontal integration*, in *Lorcan Dempsey's weblog: on libraries, services and networks*, 20 Nov. 2005, <<http://orweblog.oclc.org/archives/000865.html>>.

Dempsey, Lorcan [2009] *Untangling the library systems environment*, in *Lorcan Dempsey's weblog: on libraries, services and networks*, 25 Oct. 2009, <<http://orweblog.oclc.org/archives/002015.html>>.

De Robbio, Antonella [2002] *Open Archive. Per una comunicazione scientifica 'free online'*, "Bibliotime", anno V (luglio 2002), n. 2, <<http://didattica.spbo.unibo.it/bibliotime/num-v-2/derobbio.htm>>.

Di Donato, Francesca [2009] *Introduzione all'open access: storia, filosofia, strumenti, Che cos'è l'Open Access*, slide presentate al convegno *Un confronto tra ricercatori, bibliotecari e studenti Università Ca' Foscari*, Venezia, 3 Apr. 2009, <http://bfp.sp.unipi.it/~didonato/ovre/venezia_3aprile2009.pdf>.

Dietz, Roland – Grant, Carl [2005] *The Dis-Integrating World of Library Automation*, "Library Journal", Jun. 2005, <<http://www.libraryjournal.com/article/CA606392.html>>.

Di Giammarco, Fabio [2004] *E ora in biblioteca si affaccia RFID*, "Biblioteche oggi", 22 (2004), n. 4, pp. 73-74, <www.bibliotecheoggi.it/2004/20040407301.pdf>.

Di Giammarco, Fabio [2005] *Open source e biblioteche*, "Biblioteche oggi", 23 (2005), n. 10, pp. 68-69.

Dione, Bernard – Savard, Rejean [2008] *Managing technologies and automated library systems in developing countries: open source vs commercial options*, *proceedings of the IFLA Pre-Conference satellite meeting, Dakar, Senegal, August 15-16*, IFLA Publications n. 132,

edited by Bernard Dione and Rejean Savard, Munich, K.G. Saur, 2008.

DLF ILS-DI [2008] *DLF ILS Discovery Interface Task Group (ILS-DI) Technical Recommendation Revision 1.1*, Digital Library Federation Task Group, 8 Dec. 2008, <http://www.diglib.org/architectures/ilsdi/DLF_ILS_Discovery_1.1.pdf>.

Dougherty, William C. [2009] *Integrated Library Systems: Where Are They Going? Where Are We Going?*, "The Journal of Academic Librarianship", 35 (2009), n. 5, pp. 482-485.

Foglieni, Ornella [1984] *Il Servizio Bibliotecario Nazionale, Ipotesi di realizzazione in Lombardia, atti del seminario realizzato in collaborazione con l'Associazione italiana biblioteche - Comitato regionale lombardo*, Milano, 18-19 novembre 1983, Milano, Editrice bibliografica, 1984.

Frigimelica, Giovanna [2009] *La diffusione di software open source per la gestione di biblioteche in Italia*, presentazione al convegno "Generazione LIS 2009", Milano, 12 Mar. 2009, <<http://conferences.aepic.it/index.php/lis/2009/paper/view/32/25>>.

Frigimelica, Giovanna – Marchitelli, Andrea [2009], *Open Source in biblioteca: how to?*, "Bollettino del CILEA", 114 (Dic. 2009), pp. 38-40, <<http://bollettino.cilea.it/include/getdoc.php?id=2190&article=786&mode=pdf>>.

Gambari, Stefano [2007] *Biblioteca digitale*, in *Biblioteconomia. Guida classificata*, diretta da Mauro Guerrini. Milano, Editrice Bibliografica, 2007, pp. 309-323.

Gambari, Stefano – Guerrini, Mauro [2002] *Definire e catalogare le risorse elettroniche: un'introduzione a ISBD(ER), AACR2 e metadati*, saggio introduttivo di Paul Gabriele Weston, Milano, Editrice Bibliografica, 2002.

Goldstein, Charles M. [1983] *Integrated Library Systems*, "Bulletin of the Medical Library Association", 71 (Jul. 1983), n. 3, pp. 308-311, <<http://www.ncbi.nlm.nih.gov/pmc/articles/PMC227197/>>.

Gordon, Rachel Singer - West, Jessamyn [2008] *What Can Open Source Do for You?*, "Computers in Libraries", 28 (Mar. 2008), n. 3, pp. 44-45.

Guerra, Luca – Pelizzari, Eugenio [2001] *Approval plan in EDI: il futuro delle acquisizioni in biblioteca?*, "Bollettino AIB", 41 (2001), n. 1, pp. 21-29, <<http://www.aib.it/aib/boll/2001/01-1-021.htm>>.

Haravu, L.J. [2004] *Library automation: design, principles and practice*, New Dehli, Allied Publishers Private Limited, 2004.

Herzog, Brian [2008] *Using Firefox On Our Public Computers*, in "Swiss Army

Librarian", 8 May 2008, <<http://www.swissarmylibrarian.net/2008/05/08/using-firefox-on-our-public-computers>>.

Hodgson, Cynthia [2002] *The RFP writer's guide to standards for library systems*, Bethesda, NISO Press, 2002.

ICCU [2006] *Evoluzione dell'Indice SBN*, pagine a cura di Gabriella Contardi e Cristina Magliano, ultimo aggiornamento 15 Dic. 2006, <<http://www.iccu.sbn.it/genera.jsp?s=59&l=it>>.

ICCU [2009] *Comitato Metadati Amministrativi Gestionali (MAG)*, Comitato MAG, ultimo aggiornamento 3 Ago. 2009, <<http://www.iccu.sbn.it/genera.jsp?id=99&l=it>>.

ICCU [2010] *SBN UNIX in architettura client server*, pagine a cura di Gabriella Contardi, ultimo aggiornamento 9 Feb. 2010, <<http://www.iccu.sbn.it/genera.jsp?id=159>>.

INFONOMICS – BERLECON [2002] *Free/Libre and Open Source Software: Survey and Study*, International Institute of Infonomics - Berlecon Research GmbH, Jun. 2002, <<http://flossproject.org/report/index.htm>>.

Lagoze, Carl – Van De Sompel, Herbert [2000] *The Santa Fe Convention of the Open Archives Initiative*, "D-Lib Magazine", 6 (2000), n. 2, <<http://www.dlib.org/dlib/february00/vandesompel-oai/02vandesompel-oai.html>>.

Lankes, David R. - Silverstein, Joanne – Nicholson, Scott [2007] *Le reti partecipative, la biblioteca come conversazione*, traduzione a cura di Angela Di Iorio e Marialaura Vignocchi del Gruppo di studio sulle biblioteche digitali dell'AIB, in "AIB-WEB", Nov. 2007, <<http://www.aib.it/aib/cg/gbdigd07.htm3>>.

Leombroni, Claudio [2002] *Il Servizio bibliotecario nazionale. Idee, passioni, storia*, in *Storia delle biblioteche in Italia, Dall'Unità a oggi*, a cura di Paolo Traniello, Bologna, il Mulino, 2002, pp. 371-430.

Leombroni, Claudio [2003] *Una vicenda controversa: l'automazione delle biblioteche in Italia* in *La storia delle biblioteche: temi, esperienze di ricerca, problemi storiografici*, Roma, Associazione Italiana Biblioteche, 2003, pp. 167-196.

Leombroni, Claudio [2007] *L'automazione delle biblioteche in Italia* in *Biblioteconomia. Guida classificata*, diretta da Mauro Guerrini. Milano, Editrice Bibliografica, 2007, pp. 265-268.

Li Aiguo [2009] *Investigation of ILS in Chinese main academic libraries*, "Library Collections, Acquisitions, and Technical Services", 32 (2008), n. 3-4, pp. 115-120.

- Library Success (2009) *Open source software*, in *Library Success: A Best Practices Wiki*, last update 5 Dec. 2009, <http://www.libsuccess.org/index.php?title=Open_Source_Software>.
- Lugg Rick – Fischer Ruth [2003] *The real cost of ILS ownership*, White Paper Presented at IUG, 2003, <http://www.r2consulting.org/pdfs/ILS_Ownership_final.pdf>.
- Lynch, Clifford [2000] *From Automation to Transformation: Forty Years of Libraries and Information Technology in Higher Education*, “Educause Review”, 35 (Jan./Feb. 2000), n. 1, pp. 60-68, <<http://net.educause.edu/apps/er/erm00/pp060068.pdf>>.
- Maisonneuve, Marc [2009] *Bibliothèques, l'équipement informatique en 2008*, “Livres Hebdo”, 767 (2009), pp. 74-80, <<http://sites.google.com/site/toscaconsultants/etude-du-marche>>.
- Marchitelli, Andrea [2008a] *Le biblioteca nella percezione degli utenti. I risultati di tre indagini di OCLC*, “AIB notizie”, 20 (2008), n. 4, pp. 13-14, <<http://www.aib.it/aib/editoria/n20/0413.htm3>>.
- Marchitelli, Andrea [2008b] *Web e Library 2.0: Resoconto molto parziale di una conversazione iniziata a Venezia*, “Bollettino AIB”, 48 (2008), n. 8, pp. 383-389, <<http://www.aib.it/aib/boll/2008/0804383.htm>>.
- Morgan, Eric Lease [2004a] *An Introduction to the Search/Retrieve URL Service (SRU)*, “Ariadne”, 40 (2004), <<http://www.ariadne.ac.uk/issue40/morgan/>>.
- Morgan, Eric Lease [2004b] *Open source software in libraries*, pre-print in inglese dell'articolo pubblicato in francese su “Logiciels libres et bibliothèques”, BiblioAcid 1(2-3), May/Jun. 2004, pp. 1-8, <<http://infomotions.com/musings/biblioacid/>>.
- Morgan, Eric Lease [2004c] *Gift cultures, librarianship, and open source software development*, in *Infomotions' Musings on Information and Librarianship*, last update 14 Nov. 2004, <<http://infomotions.com/musings/gift-cultures/>>.
- Morgan, Eric Lease [2006] *A “Next generation” library catalog*, in *Lita Blog*, Jul. 2006, <<http://www.library.nd.edu/daiad/morgan/musings/ngc/>>.
- Mercado, Andrea [2007], *Hacking Firefox at my library: instructions & stuff*, in “LibraryTechtonics”, 20 Dec. 2007, <<http://librarytechtonics.info/bits/541/hacking-firefox-at-my-library-instructions-stuff/>>.
- Network Development and MARC Standards Office [2009] *MARC21 Formats*, The Library of Congress, last update 5 Dec. 2009, <<http://www.loc.gov/marc/marcdocz.html>>.

- Neville Shelley - Riding Ed [2004] *Standards: Perspectives of an Integrated Library Systems Vendor*, "Serials review", 30 (2004), n. 1, pp. 10–14.
- McGee Rob [2007] *An information Technology Strategic Planning Approach To Upgrading/Enhancing/Replacing The Library's Integrated Library System (ILS)*, presentazione al "Symposium on the Future of Integrated Library Systems", 13-15 Set. 2007, Champaign,
<<http://www.lincolntrail.info/ilssymposium2007/agendapage.html>>.
- Notess, Greg R. [2006] *The Terrible Twos: Web 2.0, Library 2.0, and More*, "Online", 30 (2006), n. 3, pp. 40-42.
- O'Reilly, Tim [2005] *What Is Web 2.0*, "O'Reilly Media", 30 Sep. 2005,
<<http://oreilly.com/web2/archive/what-is-web-20.html>>.
- OAI [2006] *Open Archives Initiative: About OAI*, Open Archives Initiative, 2006,
<<http://www.openarchives.org/OAI/OAI-organization.php>>.
- OCLC [2009] *About OCLC: Cooperation*, OCLC The world's libraries. Connected, last modified 25 Mar. 2009, <<http://www.oclc.org/us/en/about/cooperation/default.htm>>.
- OLE Project [2009] *The Open Library Environment Project Final Report*, The Open Library Environment Project, 26 Jul. 2009, <http://oleproject.org/wp-content/uploads/2009/07/ole_report_draft_26july09.pdf>.
- Pace, Andrew K. [2004], *Dismantling Integrated Library Systems*, "Library Journal", Feb. 2004,
<<http://www.libraryjournal.com/article/CA374953.html>>.
- Pace, Andrew K. [2005] *Helping You Buy: Integrated Library Systems*, "Computers in Libraries", 25 (Sep. 2005), n.8, p. 25.
- Paskin, Norman [2009] *About the International DOI Foundation*, International DOI Foundation, ultimo aggiornamento 17 Set. 2009,
<<http://www.doi.org/welcome.html>>.
- Pasqui, Valdo [2006] *Portali: funzionalità e tecnologie nel contesto bibliotecario*, dispensa della lezione tenuta presso l'Università di Parma, 5 Dic. 2006,
<<http://hdl.handle.net/1889/509>>.
- Pasqui, Valdo [2009] *Evoluzione dei sistemi di gestione bibliotecaria tra vecchi e nuovi paradigmi*, "Bollettino AIB", 49 (2009), n. 3, pp. 289-305,
<<http://www.aib.it/aib/boll/2009/0903289.htm>>.
- Peruginelli, Susanna – Giordano, Tommaso – Pettenati, Corrado [1989] *Biblioteche*

pubbliche e automazione: Un progetto per l'applicazione di SBN alle biblioteche della provincia di Modena, Milano, Editrice Bibliografica, 1989.

Pettenati, Corrado [1985] *L'automazione nelle piccole biblioteche. Uno studio di fattibilità per l'area fiorentina*, "Biblioteche oggi", 3 (1985), n. 5-6, pp. 37-61.

Picerni, Antonio - De Rossi, Alessandro [2009] *L'offerta Open Source in Italia: analisi di un settore in evoluzione*, VIU Papers 01.09,
<http://www.viucommunity.net/index.php?option=com_content&view=article&id=53>.

Potorti, Francesco [2002] *Cos'è il software libero?*, "Bibliotime", anno V (Lug. 2002), n. 2, <<http://didattica.spbo.unibo.it/bibliotime/num-v-2/potorti.htm>>.

Ransom, Joann [2009] *How Hard Can It Be? : Developing in Open Source*, with Chris Cormack and Rosalie Blake, "The Code4Lib Journal", Issue 7, 26 Giu. 2009,
<<http://journal.code4lib.org/articles/1638>>.

Raymond, Eric S. [2000] *The Cathedral and the Bazaar*, 11 Sep. 2000,
<<http://www.catb.org/~esr/writings/cathedral-bazaar/>>.

Ridi, Riccardo [2007] *La biblioteca come ipertesto*, Milano, Editrice Bibliografica, 2007.

Riewe, Linda M. [2008] *Survey of open source integrated library systems*, The Faculty of the School of Library and Information Science, San José State University, Aug. 2008,
<<http://users.sfo.com/~lmr/ils-survey/080831-paper-Riewe.pdf>>.

Ryan, Terry [2004] *Turning Patrons into Partners When Choosing an Integrated Library System*, "Computers in Libraries", 24 (Mar. 2004), n. 3, pp. 6-8,
<<http://www.infoday.com/cilmag/mar04/ryan.shtml>>.

Sadeh, Tamar [2006] *Google Scholar Versus Metasearch Systems*, "High energy physics libraries webzine", 12 (Feb. 2006),
<<http://library.web.cern.ch/library/Webzine/12/papers/1/>>.

Saffady, William [1983] *Introduction to Automation for Librarians*, Chicago, American Library Association, 1983.

Saffady, William [1989] *Library Automation: An Overview*, "Library Trends", 37 (Winter 1989), n. 3, pp. 269-281.

Schneider, Karen G. [2006] *How OPACs Suck*, ALA TechSource Blog, Mar. 13, (2006),
<<http://www.alatechsource.org/blog/2006/03/how-opacs-suck-part-1-relevance-rank-or-the-lack-of-it.html>>.

- Scolari, Antonio [2000] *UNIMARC*, Roma, Associazione Italiana Biblioteche, 2000.
- Scott, Dan [2010] *In which I perceive that gossip is not science*, in *Coffee|Code : Dan Scott*, 24 Jan. 2010, <<http://coffeecode.net/archives/209-In-which-I-perceive-that-gossip-is-not-science.html>>.
- Solimine, Giovanni - Weston, Paul G. - Fasella, Caterina [1994] *Criteri di valutazione dei sistemi di automazione per biblioteca*, Quaderni del Laboratorio Management, Firenze, IFNIA, 1994.
- Solimine, Giovanni [1995] *Controllo bibliografico universale*, Roma, Associazione Italiana Biblioteche, 1995.
- Spinello, Annalisa [2007] *Ruoli e competenze del system librarian, library manager e system manager. Primi risultati di un'indagine in Italia*, in Congressi e Seminari MAIS, 15. Feb. 2007, <<http://hdl.handle.net/1889/519>>.
- SRU Editorial Board [2008] *Protocol Transport (SRU Version 1.2 Specifications)*, The Library of Congress, last update 19 Feb. 2008, <<http://www.loc.gov/standards/sru/specs/transport.html#soap>>.
- Stallman, Richard [2010a] *Selling Free Software*, GNU Project – Free Software Foundation (FSF), last update 07 Jan. 2010, <<http://www.gnu.org/philosophy/selling.html>>.
- Stallman, Richard [2010b] *Why Open Source misses the point of Free Software*, GNU Project – Free Software Foundation (FSF), last update 31 Gen. 2010, <<http://www.gnu.org/philosophy/open-source-misses-the-point.html>>.
- Standerson, Robert [2005] *SRW/U with OAI*, “D-Lib Magazine”, 11 (Feb. 2005), n. 2, <<http://www.dlib.org/dlib/february05/sanderson/02sanderson.html>>.
- Sturman, Robi [2004] *Il software open source per la gestione integrata delle biblioteche: una nuova risorsa?*, “Bollettino AIB”, 44 (2004), n. 3, pp. 257-270, <<http://www.aib.it/aib/boll/2004/0403257.htm>>.
- Tajoli, Zeno [2005] *DOI: uno strumento per costruire la biblioteca digitale*, “Bollettino AIB”, 45 (2005), n. 1, pp. 75-83.
- Tammaro, Anna Maria [2006] *Biblioteche digitali in Italia: scenari, utenti, staff e sistemi informativi*, rapporto di sintesi del progetto *Digital Libraries Applications*, coordinato e curato da A. M. Tammaro in collaborazione con Stefano Casati e Damiana Luzzi, Firenze, Fondazione Rinascimento digitale, 2006, <<http://www.rinascimento-digitale.it/documenti-bibliotedigitaliitalia.phtml>>.

Tennant, Roy [2007] *Digital Libraries: Demise of the Local Catalog*, "Library Journal", Jul. 2007, <<http://www.libraryjournal.com/article/CA6457238.html>>.

UNICODE [2008] *Cos'è Unicode?*, traduzione italiana di M. Cimarosti, ultimo aggiornamento 8 Ago. 2008, <<http://www.unicode.org/standard/translations/italian.html>>.

UNIVE-SBA [2009a] *Presentazione*, Sistema Biblioteche, Università Ca' Foscari Venezia, ultimo aggiornamento 19 Gen. 2009, <http://www.unive.it/nqcontent.cfm?a_id=31453>.

UNIVE-SBA [2009b] *Progetto Multiricerca*, Sistema Biblioteche, Università Ca' Foscari Venezia, ultimo aggiornamento 20 Apr. 2009, <http://www.unive.it/nqcontent.cfm?a_id=38801>.

UNIVE-SBA [2009c] *Gestione di GEAC Advance*, Sistema Biblioteche, Università Ca' Foscari Venezia, ultimo aggiornamento 17 Lug. 2009, <http://www.unive.it/nqcontent.cfm?a_id=66958>.

UNIVE-SBA [2009d] *Riattivazione delle funzioni di amministrazione del software GEAC Advance*, Sistema Biblioteche, Università Ca' Foscari Venezia, ultimo aggiornamento 17 Lug. 2009, <http://www.unive.it/nqcontent.cfm?a_id=38800>.

Van den Boomen [2007] *What is Web 2.0?*, in *Meta BlogNote*, 22 Apr. 2007, <<http://metamapping.net/blog/?p=85>>.

Vitiello, Giuseppe [2004] *L'identificazione degli identificatori*, "Biblioteche oggi", 22 (2004), n. 2, pp. 67-80.

Wang, Zhonghong [2009] *Integrated Library Systems (ILS) Challenges and Opportunities: A Survey of U.S. Academic Libraries with Migration Projects*, "The Journal of Academic Librarianship", 35 (2009), n. 3, pp. 207-220.

West, Jessamyn [2009] *Ten Open Source Tools*, "Computers in Libraries", 29 (Feb. 2009), n. 2, pp. 44-45.

Weston, Paul Gabriele [2007] *La gestione elettronica delle biblioteche*, in *Biblioteconomia: principi e questioni* a cura di Giovanni Solimine e Paul Gabriele Weston, Roma, Carocci, 2007, pp. 221-256.

WIKIPEDIA [2009] *DHTML*, Wikipedia. L'enciclopedia libera, ultimo aggiornamento 27 Ago. 2009 <<http://it.wikipedia.org/wiki/DHTML>>.

WIKIPEDIA [2010a] *AJAX*, Wikipedia. L'enciclopedia libera, ultimo aggiornamento 18 Feb. 2010, <<http://it.wikipedia.org/wiki/AJAX>>.

WIKIPEDIA [2010b] *Electronic Data Interchange*, Wikipedia. L'enciclopedia libera, ultimo aggiornamento 22 Feb. 2010, <http://en.wikipedia.org/wiki/Electronic_Data_Interchange>.

WIKIPEDIA [2010c] *Fear, uncertainty and doubt*, Wikipedia. L'enciclopedia libera, ultimo aggiornamento 23 Feb. 2010, <http://it.wikipedia.org/wiki/Fear,_uncertainty_and_doubt>.

WS Arch-WG [2004] *Web Services Glossary*, W3C Web Services Architecture Working Group, W3C, 11 Feb. 2004, <<http://www.w3.org/TR/ws-gloss/>>.

Siti web di riferimento^{*}

AIB-WEB (Associazione Italiana Biblioteche) <<http://www.aib.it/>>

CNIPA – Osservatorio Open Source <<http://www.ossipa.cnipa.it/home/>>

code4lib | coders for libraries, libraries for coders <<http://code4lib.org/>>

CPAN - Comprehensive Perl Archive Network <<http://www.cpan.org/>>

Debian – The Universal Operating System <<http://www.debian.org/>>

Git – Fast Version Control System <<http://git-scm.com/>>

GNU Operating System <<http://www.gnu.org/>>

ICCU - Istituto Centrale per il Catalogo Unico delle Biblioteche Italiane e per le Informazioni Bibliografiche <<http://www.iccu.sbn.it/>>

Index Data <<http://www.indexdata.com/>>

ISO – International Organization for Standardization
<<http://www.iso.org/iso/home.html>>

Koha Library Software Community <<http://koha-community.org/>>

Koha.it: La comunità italiana di Koha <<http://www.koha.it/>>

Library Success: A Best Practices Wiki <<http://www.libsuccess.org/>>

Library Technology Guides: Key resources in the field of Library Automation
<<http://www.librarytechnology.org/>>

MySQL :: The world's most popular open source database <<http://www.mysql.com/>>

^{*} Gli indirizzi URL riportati sono stati tutti controllati l'ultima volta in data 5 marzo 2010.

NISO - National Information Standards Organization <<http://www.niso.org/home/>>

Open Source Initiative <<http://www.opensource.org/>>

oss4lib | open source systems for libraries <<http://www.oss4lib.org/>>

Perl Monks – The Monastery Gates <<http://www.perlmonks.org/>>

SourceForce.net: Find and Develop Open Source Software <<http://sourceforge.net/>>

Standards at the Library of Congress <<http://www.loc.gov/standards/>>

The Perl Programming Language <<http://www.perl.org/>>

UML® Resource Page <<http://www.uml.org/>>

World Wide Web Consortium (W3C) <<http://www.w3.org/>>

Ringraziamenti

A mia mamma e mio papà, a Mattia, Federico, Lucia, Santosh e tutta la famiglia.

A Sara. Agli amici, vicini o lontani, Luca, Matteo, Benedetta, Laura, Gianluca, Pozzo, Laura, Mattia, Nükhet, Guglielmo, Julia, Raffaele, Cristina, Franz, Federico, Francesco, Carlo, Bea, Giorgia, Enrica, Francesca, Timoteo, Akiko, Marco, Marina, Chiara, Silvia, Paola, Marco, Alessandro, Giulia, Alice, Giulia, Rocco, Giulia, Alberto, Irene, Marta, Angela, Serena, Andrea, Fernando, Martina, Nicolò, Riccardo, Valeria, Monica, Michele, Margherita, Ginevra, Marta, Gnappetta, Mario, Francesca, Barbara, Annina, Giorgio, Riccardo.

Alle mie colleghe Sandra e Vania.

Ai preziosi aiuti di Zeno Tajoli, Pierfranco Minsenti, Gianluca Dei Rossi, Giancarlo Bascone.

A tutto lo staff della Biblioteca di Area Economica, in particolare la direttrice dott.ssa Daniela Grandin, Carlo Miclet, Vito Griseta, Natascia Danieli, Ines Andreetta, Roberta Cavallarin per l'enorme disponibilità e pazienza.

A mia nonna Giovannina.